Alan Turing
padre de la Computación

Verónica Becher

Universidad de Buenos Aires

Ecos de la figura y de la obra de Turing en la Argentina
Universidad Nacional de General Sarmiento
10 de noviembre, 2017

Blackpool
Leeds
Kingston upon Hull
Great Britain
Liverpool
Sheffield
Chester
Peak District National Park
Nottingham
Snowdonia National Park
Leicester
Peterborough
Norwich
INGLATERRA
Cambridge
Ipswich
Colchester
Cotswolds AONB
Oxford
Swansea
Swindon
Londres
Cardiff
Bristol
Reading
Bath
Canterbury
Southampton
Brighton
Bournemouth
Portsmouth
Exeter
Torquay
Plymouth
Canal de la Mancha
Calais
Dunquerque
Brujas
Amberes
Gante
Bruselas
Lille
Arrás
Amiens
La Haya
Róterdam


MAIDA
VALE W9
CITY OF WESTMINSTER

Alan Turing

Alan Turing

Nacido en Maida Vale, London, 23 June 1912.

Alan Turing

Nacido en Maida Vale, London, 23 June 1912.

Hermano mayor: John.

Alan Turing

Nacido en Maida Vale, London, 23 June 1912.

Hermano mayor: John.

Papá: Julius Mathison Turing,1873–1947, posición en Indian Civil Service (ICS) at Chhatrapur, Bihar and Orissa Province, in British India.

Alan Turing

Nacido en Maida Vale, London, 23 June 1912.

Hermano mayor: John.

Papá: Julius Mathison Turing,1873–1947, posición en Indian Civil Service (ICS) at Chhatrapur, Bihar and Orissa Province, in British India.

Mamá: Ethel Sara Stoney, 1881–1976.

Abuelo paterno: John Robert Turing, clérigo, de familia esocesa de comerciantes (Holanda)

Abuelo materno: Edward Waller Stoney, ingeniero de Madras Railways. Los Stoney eran burguesía protestante anglo irlandesa.

ALAN M.

TURING

Centenary Edition

SARA TURING

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Primaria (desde los 6 los 10): St Michael's, St Leonards-on-Sea (Hastings).

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Primaria (desde los 6 los 10): St Michael's, St Leonards-on-Sea (Hastings).

Preparatoria (desde los 10 a los 13): Hazelhurst Preparatory School, in Frant, Sussex.

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Primaria (desde los 6 los 10): St Michael's, St Leonards-on-Sea (Hastings).

Preparatoria (desde los 10 a los 13): Hazelhurst Preparatory School, in Frant, Sussex.

Secundaria Sherborne School (desde los 13 a los 18).

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Primaria (desde los 6 los 10): St Michael's, St Leonards-on-Sea (Hastings).

Preparatoria (desde los 10 a los 13): Hazelhurst Preparatory School, in Frant, Sussex.

Secundaria Sherborne School (desde los 13 a los 18).
Christopher Morcom.

Educación pública en el imperio britanico

Durante la niñez de Alan Turing sus padres viajaban entre Inglaterra e India, dejando a los dos niños con una pareja de militares retirados.

Primaria (desde los 6 los 10): St Michael's, St Leonards-on-Sea (Hastings).

Preparatoria (desde los 10 a los 13): Hazelhurst Preparatory School, in Frant, Sussex.

Secundaria Sherborne School (desde los 13 a los 18).
Christopher Morcom.

El biógrafo Andrew Hodges reconstruyó esta parte de la historia.

Andrew Hodges escribió

Su hermano decía que Alan era desalineado.

Andrew Hodges escribió

Su hermano decía que Alan era desalineado.

Alan fue más incompetente con las exigencias escolares del imperio británico que rebelde.

Andrew Hodges escribió

Su hermano decía que Alan era desalineado.

Alan fue más incompetente con las exigencias escolares del imperio británico que rebelde.

Christopher Morcom le hizo notar su falta de prolijidad. Y la necesidad de rigor expositivo.

Andrew Hodges escribió

Su hermano decía que Alan era desalineado.

Alan fue más incompetente con las exigencias escolares del imperio británico que rebelde.

Christopher Morcom le hizo notar su falta de prolijidad. Y la necesidad de rigor expositivo.

Turing tenía voz de pito (alto tono) que no agradaba.

King's College Cambridge 1930-1934

Desde los 18 a los 22 Turing estudió matemática en King's College Cambridge.

Turing fue alumno de G.H. Hardy y colega de D. Champernowne.

King's College Cambridge 1930-1934

Desde los 18 a los 22 Turing estudió matemática en King's College Cambridge.

Turing fue alumno de G.H. Hardy y colega de D. Champernowne.

G.H. Hardy no llegó apreciar la contribución de Turing.
Champernowne y Turing, respeto mutuo, numeros normales, y "TuroChamp"

King's College Cambridge 1930-1934

Desde los 18 a los 22 Turing estudió matemática en King's College Cambridge.

Turing fue alumno de G.H. Hardy y colega de D. Champernowne.

G.H. Hardy no llegó apreciar la contribución de Turing.
Champernowne y Turing, respeto mutuo, numeros normales, y "TuroChamp"

En 1935 Turing fue miembro (fellow) de King-'s College Cambridge.

King's College Cambridge 1930-1934

Desde los 18 a los 22 Turing estudió matemática en King's College Cambridge.

Turing fue alumno de G.H. Hardy y colega de D. Champernowne.

G.H. Hardy no llegó apreciar la contribución de Turing.
Champernowne y Turing, respeto mutuo, numeros normales, y "TuroChamp"

En 1935 Turing fue miembro (fellow) de King-'s College Cambridge.

En 1936 se fue a Princeton a estudiar con Alonzo Church.

Proceedings of the Lodon Mathematical Society

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

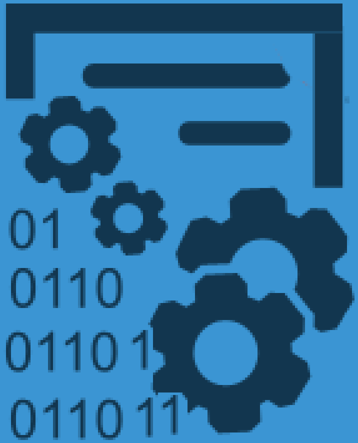
1. *Computing machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach § 9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ which will be called "*m*-configurations". The machine is supplied with a

3. *Examples of computing machines.*

I. A machine can be constructed to compute the sequence 010101 … . The machine is to have the four *m*-configurations "$\mathfrak{b}$", "$\mathfrak{c}$", "$\mathfrak{f}$", "$\mathfrak{e}$" and is capable of printing "0" and "1". The behaviour of the machine is


01
0110
01101
011011

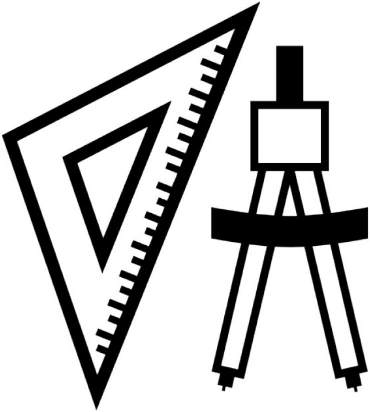
6. *The universal computing machine.*

It is possible to invent a single machine which can be used to compute any computable sequence. If this machine $\mathcal{U}$ is supplied with a tape on the beginning of which is written the S.D of some computing machine $\mathcal{M}$,

¿Por qué?

Regla y compás

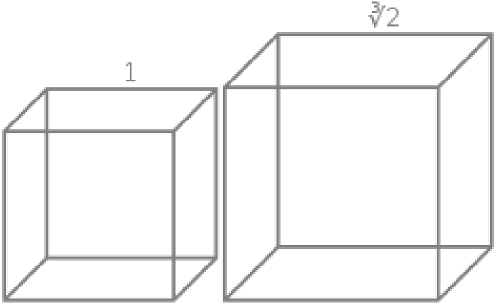
Los griegos se preguntaban qué problemas geométricos podían resolverse con regla y compas. Y qué problemas no.



La regla y el compás no alzancan

Problema

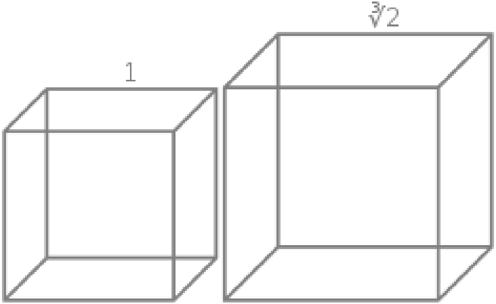
Dado un cubo, construir otro que tenga el doble de volumen.



La regla y el compás no alzancan

Problema

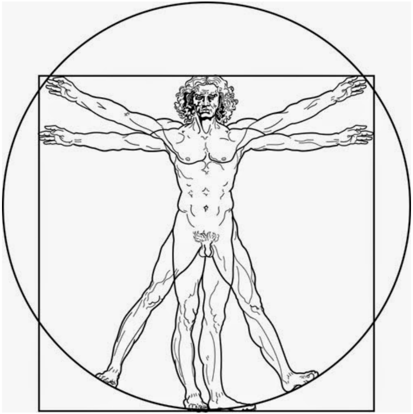
Dado un cubo, construir otro que tenga el doble de volumen.



En 1837 matemático francés Pierre Wantzel demostró que es imposible hacerlo solamente con con regla y compas.

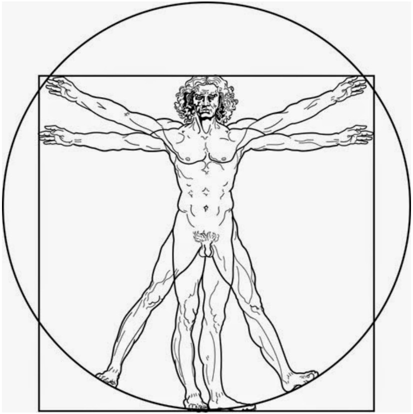
La regla y el compás no alzancan

Problema
La cuadratura del círculo.



La regla y el compás no alzancan

Problema
La cuadratura del círculo.



En 1882 el matemático alemán Ferdinand Lindemann probó que π es un número trascendente y eso implica que es imposible cuadrar el círculo usando regla y compás.

1900

Abaco para todos los enunciados matemáticos

Problema
Para cada enunciado matemático (si-no),
¿se puede determinar finitariamente si es demostrable?

Abaco para todos los enunciados matemáticos

Problema
Para cada enunciado matemático (si-no),
¿se puede determinar finitariamente si es demostrable?

Ejemplos:

Dado un numero entero x, ¿es x primo?

Abaco para todos los enunciados matemáticos

Problema
Para cada enunciado matemático (si-no),
¿se puede determinar finitariamente si es demostrable?

Ejemplos:

Dado un numero entero x, ¿es x primo?

Dados dos numeros enteros x e y, ¿es x divisible por y?

La noción de algoritmo

Abu Abdallah Muhammad Musa al-Khwarizmi (780-850) fue un matemático y astrónomo persa, a quien se le atribuye la noción algoritmo.

La noción de algoritmo

Abu Abdallah Muhammad Musa al-Khwarizmi (780-850) fue un matemático y astrónomo persa, a quien se le atribuye la noción algoritmo.

Por ejemplo, el algoritmo de Euclides (325 A. C.- 265 A.C) para calcular el máximo comun divisor.

Enstcheidungsproblem

Para cada enunciado matemático (si-no),
¿hay un algoritmo que determine si es demostrable?

1936

Enstcheidungsproblem resuelto

Hay enunciados matemáticos (si-no) tales que ningun algoritmo puede determinar si son demostrables.

Enstcheidungsproblem resuelto

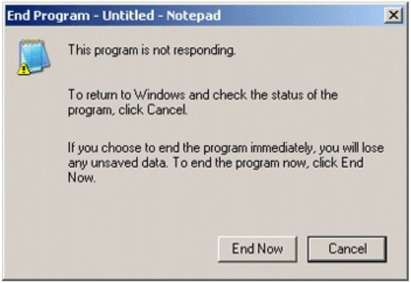
Hay enunciados matemáticos (si-no) tales que ningun algoritmo puede determinar si son demostrables.

Alan Turing. "On Computable numbers with an Application to the Entschaidungsproblem." Proceedings of London Mathematical Society, 1936.

Alonzo Church. "A note on the Entscheidungsproblem", Journal of Symbolic Logic, 1936.

La solución de Turing

Su programa no responde ¿Qué deseas hacer?



La solución de Turing

Turing define la computadora como mecanismo finitario de cálculo.
Un algoritmo es una función calculable por una computadora universal.

La solución de Turing

Turing define la computadora como mecanismo finitario de cálculo.
Un algoritmo es una función calculable por una computadora universal.

Turing formaliza el cómputo como objeto matemático. Dice lo que las computadoras pueden hacer y lo que no.

La solución de Turing

Turing define la computadora como mecanismo finitario de cálculo.
Un algoritmo es una función calculable por una computadora universal.

Turing formaliza el cómputo como objeto matemático. Dice lo que las computadoras pueden hacer y lo que no.

Turing matematizó lo que las computadoras no pueden hacer.

La solución de Turing

Problema
¿Para cada pregunta matemática si-no, hay un algoritmo que la responde?

La solución de Turing

Problema
¿Para cada pregunta matemática si-no, hay un algoritmo que la responde?

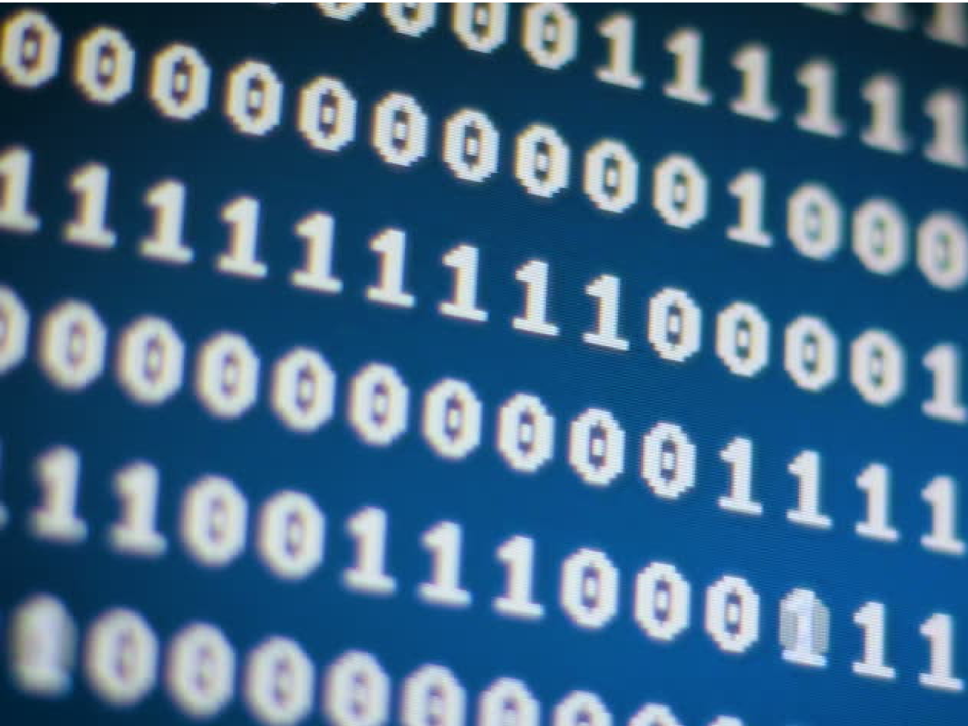
No.

La solución de Turing

Problema
¿Para cada pregunta matemática si-no, hay un algoritmo que la responde?

No. No hay ningún algoritmo capaz de determinar si otro algoritmo entra en un ciclo infinito. Es decir, ningun pograma que se ejecuta en una computadora puede determinar si otro programa está colgado o no.

Premio Turing, desde 1966



Las máquinas de Turing son las de Internet

Las máquinas de Turing son las de Internet

Turing concibe sus computadoras como dispositivos que operan en base a una configuración finita de acciones, eternamente.

Las máquinas de Turing son las de Internet

Turing concibe sus computadoras como dispositivos que operan en base a una configuración finita de acciones, eternamente.

Este es el modelo de cómputo que tenemos en la actualidad, y más allá de las consideraciones tecnológicas, el modelo de cómputo de la computadoras personales es exactaemnte el de una máquina de Turing, que opera para siempre.

Turing prolífico

teorema central del límite

Turing prolífico

teorema central del límite
números normales

Turing prolífico

teorema central del límite
números normales
ordinales

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann
criptografía

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann
criptografía
morfogénensis

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann
criptografía
morfogénensis
máquina inteligente (es una definición probabilística).

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann
criptografía
morfogénensis
máquina inteligente (es una definición probabilística).
“Turchamp” juego de ajedrez

Turing prolífico

teorema central del límite
números normales
ordinales
función zeta de Reimann
criptografía
morfogénensis
máquina inteligente (es una definición probabilística).
"Turchamp" juego de ajedrez
...

Concibió pero no formalizó

Para Turing la computadora daba un método para reducir el tiempo que lleva resolver un problema.

Concibió pero no formalizó

Para Turing la computadora daba un método para reducir el tiempo que lleva resolver un problema.

Experimentó el problema de la complejidad computacional pero no formalizó la teoría de la complejidad.

Concibió pero no formalizó

Para Turing la computadora daba un método para reducir el tiempo que lleva resolver un problema.

Experimentó el problema de la complejidad computacional pero no formalizó la teoría de la complejidad.

Más tarde surge la teoría de complejidad (tiempo, espacio, peor caso, caso promedio) que formula la pregunta “P versus NP”, aún hoy abierta.

Para Turing

Una computadora es inteligente si puede mentir (juego de la imitación).

Para Turing

Una computadora es inteligente si puede mentir (juego de la imitación).

La definición de inteligencia es probabilística: la computadora es inteligente si sus mentiras tienen la misma proporción de éxito que las mentiras de un ser humano.

Para Turing

Una computadora es inteligente si puede mentir (juego de la imitación).

La definición de inteligencia es probabilística: la computadora es inteligente si sus mentiras tienen la misma proporción de éxito que las mentiras de un ser humano.

Explícitamente Turing deja de lado el cuerpo y el movimiento.

Para Turing

Una computadora es inteligente si puede mentir (juego de la imitación).

La definición de inteligencia es probabilística: la computadora es inteligente si sus mentiras tienen la misma proporción de éxito que las mentiras de un ser humano.

Explícitamente Turing deja de lado el cuerpo y el movimiento.
No pone en juego el espacio.

¿Sorpresa?

¿Sorpresa?

Turing no vio que la computadora podía ser un método para reducir también el espacio.

¿Sorpresa?

Turing no vio que la computadora podía ser un método para reducir también el espacio. Si Turing viniera, se sorprenderia que su creación también puede reducir el espacio: Internet.