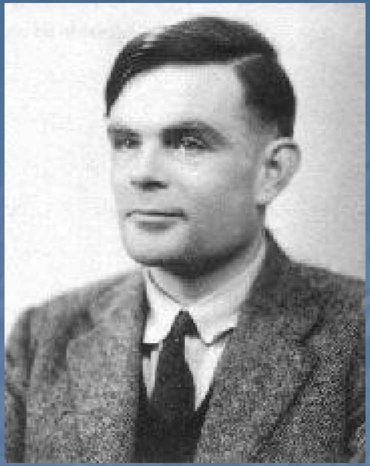
Hyper computation

Introduction & Philosophy

Preface

- Jeroen Broekhuizen
  - History before Hyper computing
- Christian Gilissen
  - Introduction & philosophy of Hyper computing
- Maurice Samulski
  - Hyper computing by examples

Alan Turing



- Well known
- Mostly known as the inventor of the *Turing Machines*
- Also invented other machines & theories

Algorithm

- Turing made Turing Machines for formalizing notion of *algorithms*
- Algorithm:
  - systematic procedure that produces – in finite number of steps – the answer to a question or the solution of a problem
- Named after the mathematician Al-Koarizmi from the 9-th century

Algorithmic computation

- Algorithmic computation:
  - The computation is performed in closed-box, transforming finite input, determined at start of the computation, to finite output in a finite amount of time.
- Matches properties of TM

Turing Machines

- Have properties that model algorithmic computation:
  - Computation is closed
  - Resources are finite (time & tape)
  - Behavior is fixed (start in same configuration)

Strong Turing Thesis

- Thesis:
  - A Turing Machine can do everything a real computer can do.
- Wrong interpretation Church-Turing Thesis
- Alan Turing would have disagreed
- Proposed other models with properties that contradict the algorithmic properties

Turing's contributions

- Entscheidungsproblem
- Turing's thesis
- Choice- and Oracle Machines
- Cryptology and complexity theory
- ACE: general universal computers
- Turing’s Unorganized Machines
- Artificial intelligence & life

Entscheidungsproblem

- What is it?
- Can you think of an example?

Entscheidungsproblem

- Decision problem proposed by David Hilbert in 1918.
- Entscheidungsproblem :
  - Any mathematical proposition can be decided (proved true of false) by mechanistic logical methods.
- Disproved by Gödel in 1931
  - Showed that for any formal theory, there will be undecidable theorems outside its reach.

Automatic Machines

- Now called Turing Machines
- Turing continued Gödel’s work
- Proved Halting-problem is undecidable

Turing's Thesis

- Busy time around 1930:
  - Gödel invented recursive functions
  - Church invented λ-calculus
  - Turing established third class of functions computable by Turing Machines
- Both Church and Turing searched for effective ways of computing

Turing’s Thesis

- Thesis:
  - Whenever there is an effective method for obtaining the values of a mathematical function, the function can be computed by a Turing Machine.
- Based on infinite length of tape

Church-Turing Thesis

- Thesis:
  - The formal notions of recursiveness, λ-definability, and Turing-computability equivalently capture the intuitive notion of effective computability of functions over integers.

Church-Turing Thesis

- Applied to functions over integers
- Easily extendable to functions over strings
- Great influence on field ‘computer science’

Choice Machines

- Alternate method for computing
- Choice machines:
  - Partially determined by configuration
  - In some configurations it stops for interaction
  - External operator has to make a choice

Oracle Machines

- Believed: formalization of the c-machine
- Similarity with c-machines:
  - Both make queries to external agent
- Formal description Oracle:
  - A set that can be queried about any *value*; it returns *true* if the query value is in this set and *false* otherwise.

Oracle Machines

- Turing excluded the possibility that the oracle was an effective computing entity:
  - "We shall not go any further into the nature of this oracle apart from saying it cannot be a machine."

(Systems of Logic based on Ordinals, Turing A.)

Cryptology & complexity theory

- Turing contributed to breaking Enigma
- Mechanized decryption process with Turing Bombe (later the Colossus)
- Pioneered an interactive randomized approach to breaking ciphers

ACE: general universal computer

- Automatic Computing Engine
- Postwar attempt for a working computer
- Turing:
  - "Machines such as the ACE may be regarded as practical versions of the Turing Machine. There is at least a very close analogy."

(Lecture to the London Math. Society on 20'th February 1947, Turing A.)

ACE: general universal computer

- Radical innovative design, unknown till named RISC
- Too revolutionary, project was put hold

(The ACE Report, Turing A.)

Turing’s Unorganized Machines

- Two types:
  - Based on *Boolean networks*
  - Based on *finite state machines*
- Blueprint for future neural networks

(Intelligent Machinery, Turing A.)

(Turing's Connectionism: An Investigation of Neural Networks Architectures, Turing A.)

Artificial intelligence & life

- Chess as starting point for search intelligent search strategies
- Turing estimated computer beats human around 1957 → in 1997 supercomputer *Deep Blue* beats Garry Kasparov

Turing Test (for AI)

- Turing:
  - If a computer, on the basis of its written responses to questions, could not be distinguished from a human respondent, then one has to say that the computer is thinking and must be intelligent.

Hilberts Tenth Problem

- Determination of the solvability of a Diophantine equation.
  - Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: *To devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*

(http://logic.pdmi.ras.ru/Hilbert10/)

Hilberts Tenth Problem

- Typical Diophantine equation:
  - $3x^2y - 7y^2z^3 = 18$
  - $-7y^2 + 8z^2 = 0$
- Proven by Yuri Matiyasevich as unsolvable

(Quantum Hypercomputing, Tien D. Kieu)

Summary

- Turing has done lots of important work
- Unfortunately not always credit
- There is more than only Turing Machines

Preface

- Jeroen Broekhuizen
  - History before hyper computing
- Christian Gilissen
  - Introduction & philosophy of Hyper computation
- Maurice Samulski
  - Hyper computing by examples

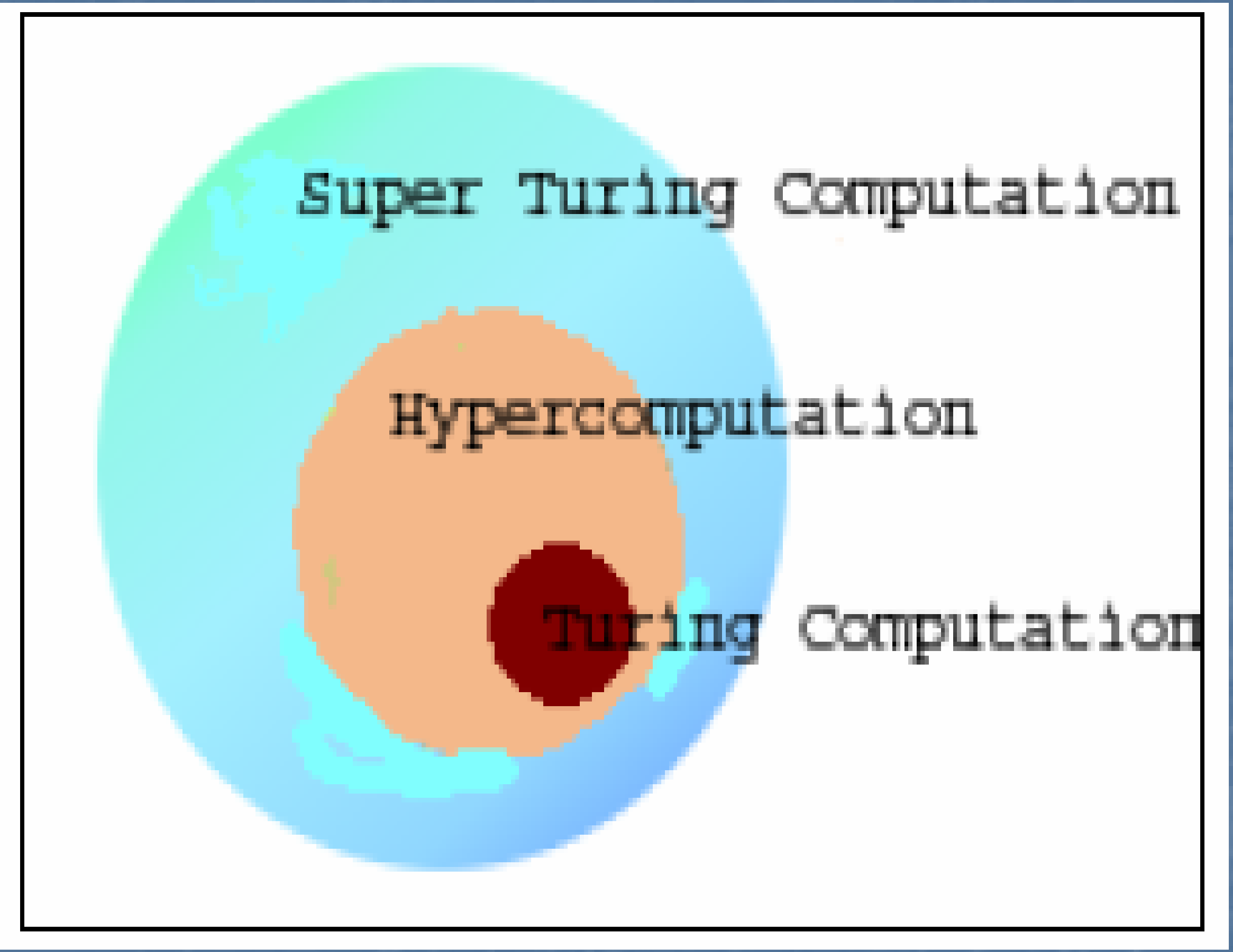
Hyper Computation

- Theoretical
- Highly discussed
- Crosses with physics and philosophy

- 3 views:
  - No HC
  - HC but not with our current laws of physics
  - HC is already implemented

Definitions

- Super-Turing: any form of information processing that a Turing machine cannot do
- Super-Turing computation, which has been used in the neural network literature to describe machines with various expanded abilities
- Hypercomputation is the theory of methods for the computation of non-recursive functions.
- Natural computation: computation occurring in, or inspired by nature


Super Turing Computation
Hypercomputation
Turing Computation

Some theses

A. All processes performable by idealized mathematicians are simulable by TMs
B. All mathematically harnessable processes of the universe are simulable by TMs
C. All physically harnessable processes of the universe are simulable by TMs
D. All processes of the universe are simulable by TMs
E. All formalisable processes are simulable by TMs

Possibilities

B C D: there is no HC in the universe.

TMs suffice to simulate all processes.

B C: The universe is HC, but no more power can be harnessed than that of a TM

B: the universe is HC, and it is at least theoretically possible to build a HC.

[none]: HC exists in the universe and is accessible

(Hypercomputation: computing more than the Turing machine, Toby Ord)

Extensions of TM's

- O-machines
- TM's with initial inscription
- Coupled TM's
- Asynchronous networks of TM's
- Error prone TM's
- Probabilistic TM's
- Infinite state TM's
- Accelerated TM's
- Fair non-deterministic TM's

Other systems

- Quantum Mechanical systems
- Analog computers
- Pulse computers

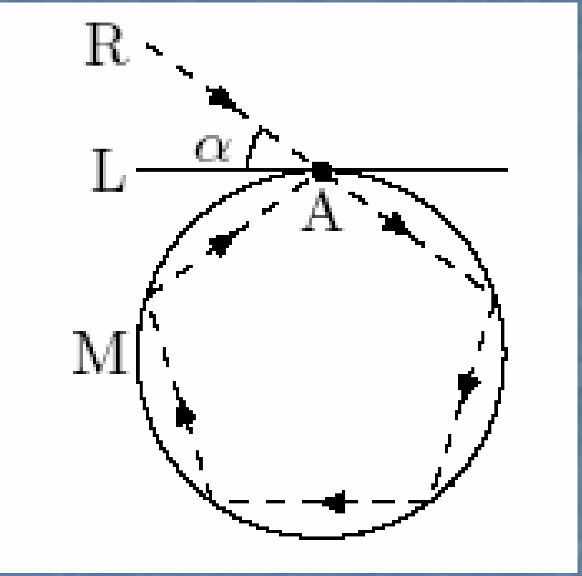
Models for TMs:

- Infinite memory
- Non-recursive information source
- Infinite specification
- Infinite time

Three views: No HC

- Most HC devices are physically impossible
  - Accelerating TM
  - Analog computers
  - Analog Neural networks

Illustration



An Illustration A simple analog apparatus capable of doing (something that no Turing machine can do (after F. Waismann 1959).

Beckenstein bound

- The Beckenstein Bound:

A spherical region with radius R and energy E can contain only a limited amount of information I

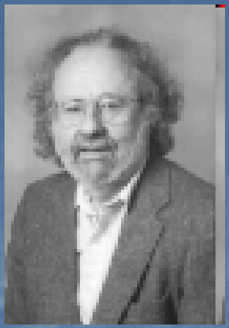
- Entails that HC is physically impossible

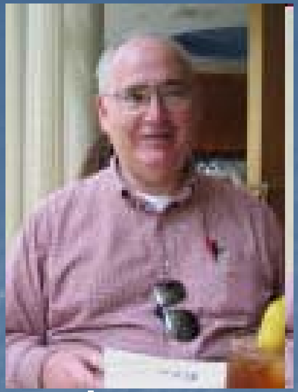
Empirical Meaningfulness

- the claim that a given device is a hypercomputer rather than a Turing is in a sense empirically meaningless.

(Hypercomputation, Gert-Jan C. Lokhorst)

(Hypercomputation: philosophical issues, B. Jack Copeland)





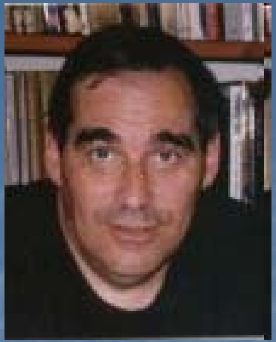
- 70 years of research on Turing degrees has shown the structure to be extremely complicated. In other words, the hierarchy of oracles is worse than any political system. No one oracle is all powerful.
- Suppose some quantum genius gave you an oracle as a black box. No finite amount of observation would tell you what it does and why it is non-recursive. Hence, there would be no way to write an algorithm to solve an understandable problem you couldn't solve before! Interpretation of oracular statements is a very fine art - as they found out at Delphi!

However

In short it would (or should) be one of the greatest astonishments of science if the activity of Mother Nature were never to stray beyond the bounds of Turing-machine-computability.

(Beyond the Universal Turing Machine, Copeland and Sylvan)

HC? Yes but not here!



- Spacetime structures in General Relativity.
  - Unlimited time
  - Unlimited space
- Hogarth showed that in M-H spacetimes either the Halting Problem or the Entscheidungsproblem can be computed by a TM.

(The physical and philosophical implications of the Church-Turing Thesis, Eleni Pagani)
(Physical Hypercomputation and the Church–Turing Thesis, ORON SHAGRIR and ITAMAR PITOWSKY)

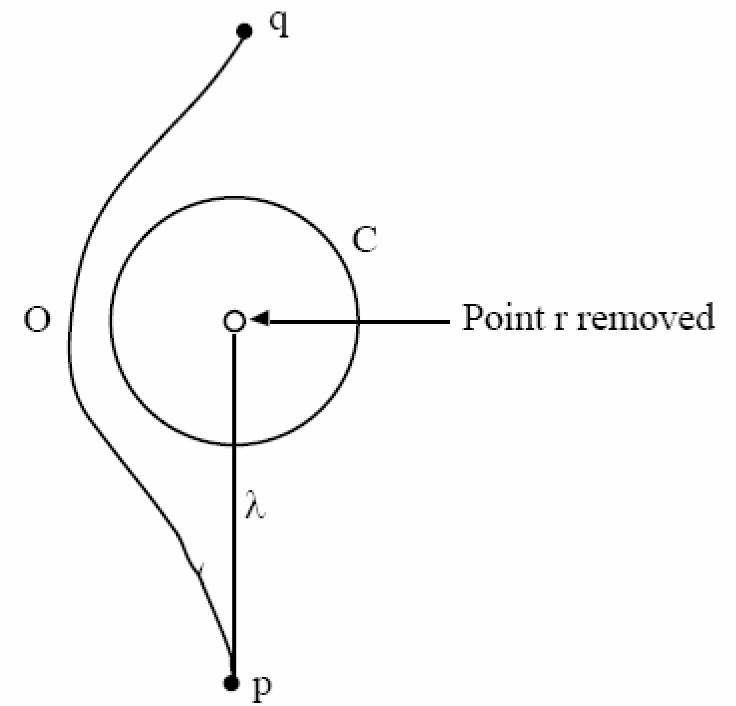


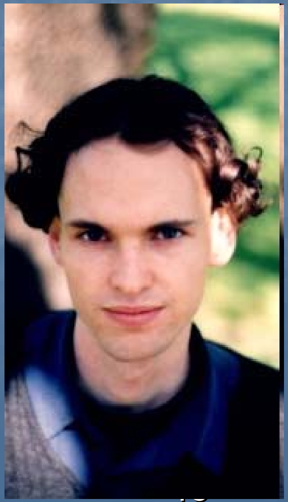
Figure 1. A toy Malament-Hogarth spacetime.

HC is already used!

- More exact: Super-Turing Computation

- Driving home from work
- Cannot be solved algorithmically but is nevertheless computable.

Hypercomputation: computing more than the Turing Machine, Toby Ord



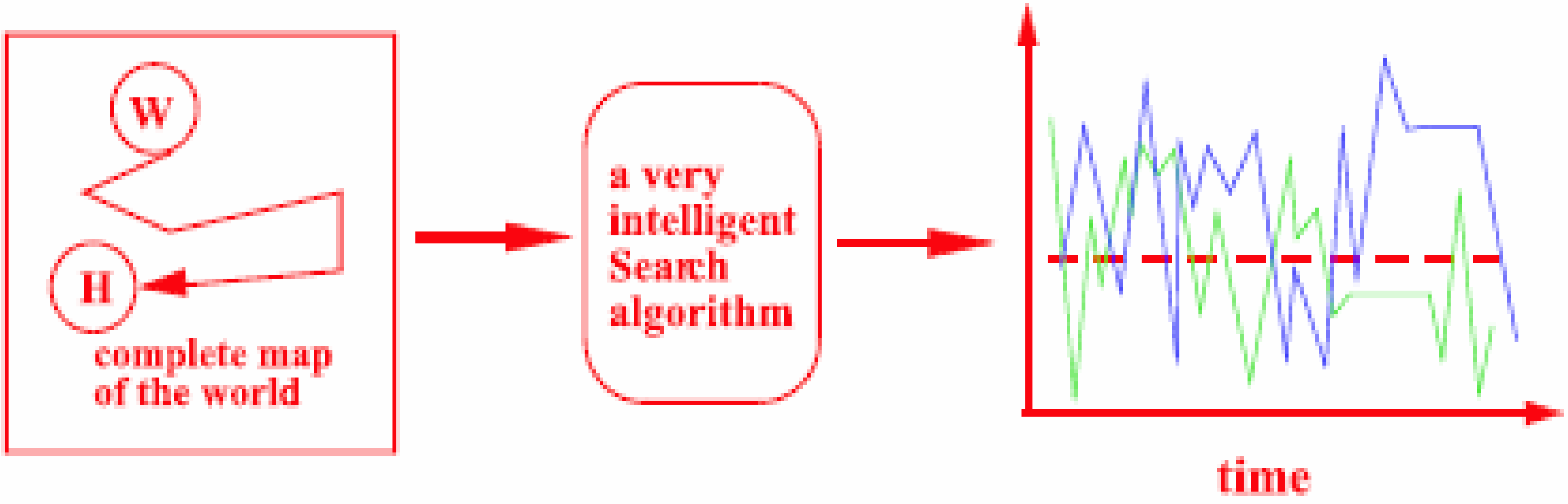


Fig. 2. Driving home from work: the algorithmic scenario

- Typical AI scenario
- Input is not precisely definable: humans

- So computational tasks situated in the real world which includes human agents are not solvable algorithmically

- Nevertheless it is computable:
- We use a driving agent that percepts on-line

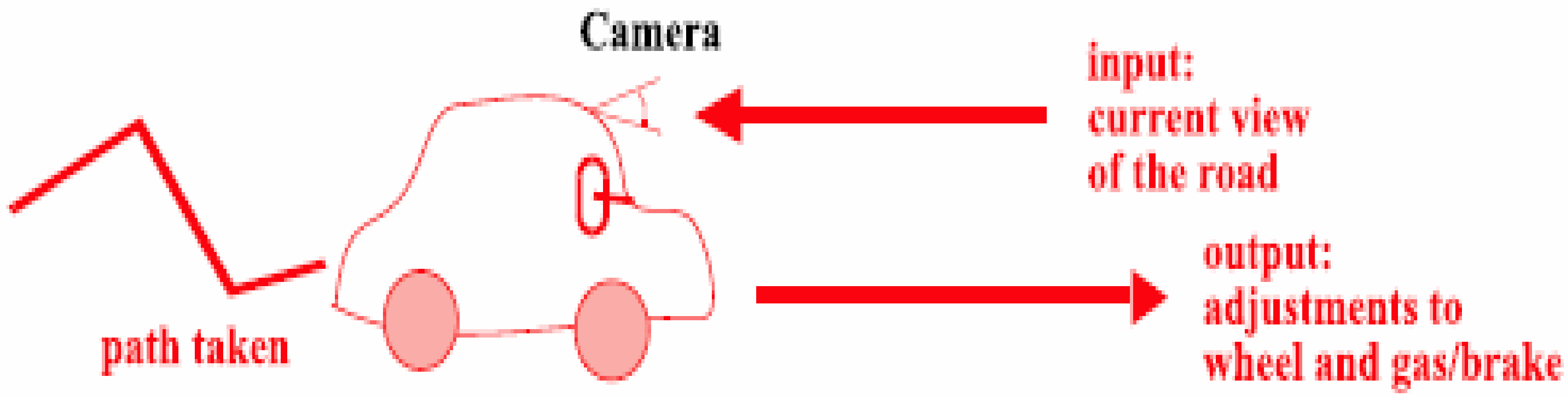


Fig. 3. Driving home from work: the interactive scenario

Real-life examples

- Distributed Client/Server computation
- Mobile robotics
- Evolutionary computation

In summary

- Almost everybody believes it exists
- But no one really knows whether it is harnessable

Preface

- Jeroen Broekhuizen
  - History before super-computing
- Christian Gilissen
  - Introduction & philosophy of Super-TM’s
- Maurice Samulski
  - Hyper computing by examples

Preface

- Jeroen Broekhuizen
  - History before super-computing
- Christian Gilissen
  - Introduction & philosophy of Super-TM’s
- Maurice Samulski
  - Hyper computing by examples

Outline

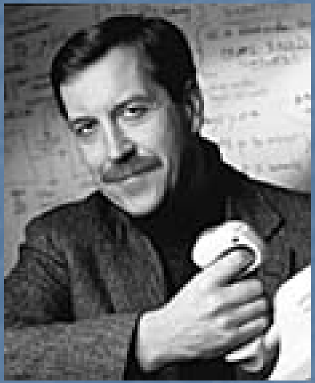
1. Do Humans Hypercompute?

2. Can computers think?

Do Humans Hypercompute?

1. Mathematicians do infinitary reasoning
2. Kinds of visual processing
3. We seem to be able to solve the halting problem

Is human cognition non-computable?



- Maybe. How about free will?
- For example, we seem to be able to generate truly random numbers
- Prof. Bringsjord claims that not all of human reasoning is computation because of our capacity to generate random numbers

Is human cognition non-computable?

- How about Infinitary Reasoning?

- Aristoteles makes distinction between
  1) "potential" infinity
  2) "actual" infinity

Experiment

- Rensselaer Polytechnic Institute
- Observe Free Will and Infinitary Reasoning
- Test ability to exhibit randomness.
- Test ability to visualize infinite

Information on sample

- Test administered to 31 students of the Rensselaer Polytechnic Institute
- Primarily first year computer science and engineering.

Random Number Generation

- Test subject generates number between 1274862 and 1972335

Character String Generation

- Subject is asked to imagine flipping a coin 20 times.
- Subject is asked to write T for tails and H for heads.

Results: Random Numbers

Digit	0	1	2	3	4	5	6	7	8	9
Frequency	19	7	18	15	19	12	13	18	17	12

	Expected Value	Actual Value
Repeats	12	20
High-Low	50	48
Low-High	50	52
Alternations	42	43

- One definition of randomness implies that the frequency of the digits should be the same

Test results: Coin Toss

Set 1	Expected	Actual	Set 3	Expected	Actual
Repeats	72	57	Repeats	72	73
High-Low	36	48	High-Low	36	41
Low-High	36	47	Low-High	36	38
Alternations	36	55	Alternations	36	34
Set 2	Expected	Actual	Set 4	Expected	Actual
Repeats	72	58	Repeats	63	52
High-Low	36	49	High-Low	32	40
Low-High	36	45	Low-High	32	41
Alternations	36	53	Alternations	32	44

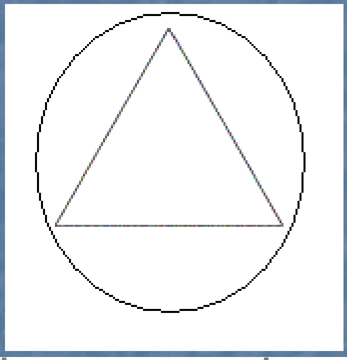
- a “high-low” corresponds to a tails followed by a heads
- 25 of 31 subjects began their strings with tails.
- Of 620 total events, 140 are heads, 480 are tails.

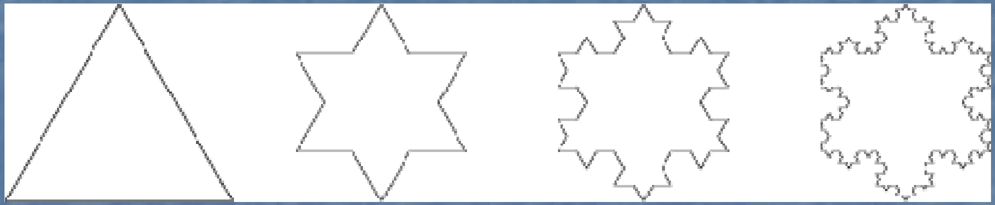
Achilles Runner

- A runner runs for 1/2 minute, then rests for 1/2 minute, then runs again for 1/4 minute, then rests for 1/4 minute, and so on.
- Test subject is asked how many times the runner will have stopped and started in two minutes.
- This represents an infinite mathematical series
- 25 students gave the correct answer, 6 were false

Koch Curve (or "Snowflake")

- Suppose that you draw a triangle inside a circle
- Now, add a new triangle 1/3 the size of the original at each side of the original





- After repeating these steps an infinite amount of times, what will the perimeter be of the last shape your draw? Will this shape fill the circle?

Koch Curve (or "Snowflake")

- The answer should be that the perimeter is infinite and that the shape will not fill the circle
- The first question was answered correct by 9 people, 22 people were incorrect
- The second question was answered correct by 7 people, 24 were incorrect

Their conclusions

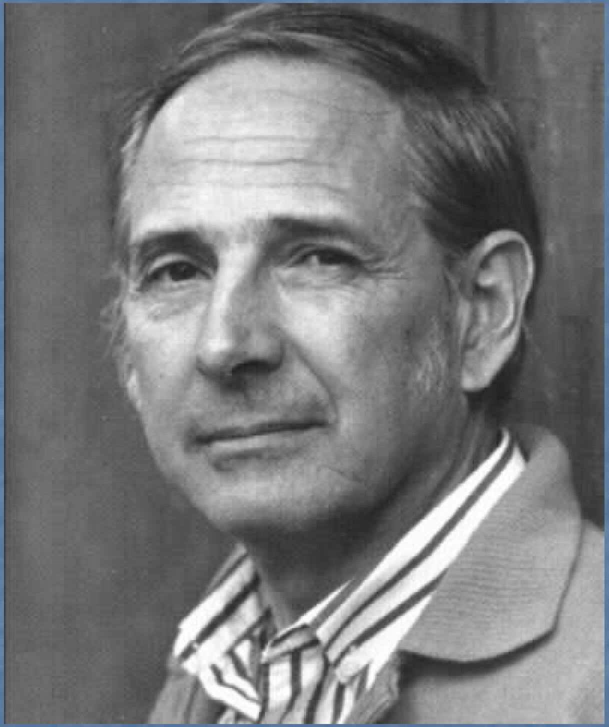
- Unlikely that humans generate truly random numbers. Perhaps we have sophisticated pseudo-random number generation algorithms, but it is not obvious that we have the ability to generate truly random numbers.
- Success with infinitary reasoning is inconsistent at best. It is not obvious that the test subjects have used any capacity for infinitary reasoning to make conclusions about the convergence of the fractals. Correct solutions could just as easily be attributed to previous knowledge or experience.

Can computers think?

- Imitation Game - Turing Test
- 3 participants: interrogator, a human and a machine
- Questions like: What dream did you have last night?
- Turing prediction: year 2000 at least 70% success

The Chinese Room Argument

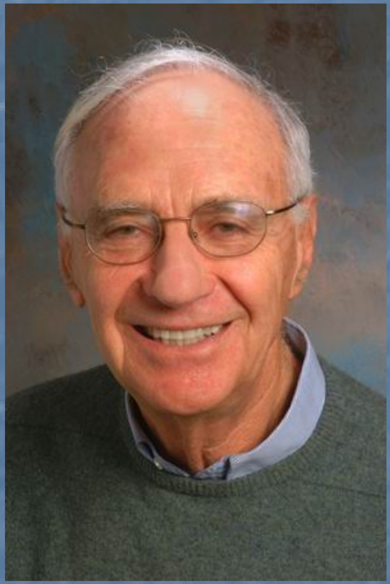
- Thought experiment designed by John Searle 1980
- Searle beliefs that such a system could indeed pass a Turing Test



Chinese Room

Chinese Room Objection

- Peter Kugel
- There is no understanding in the room because its computer imitation is too weak
- If we allowed the book to write on itself, it could "remember" and it could change what it does as a result of what it "experiences"
- This would achieve intentionality which is exactly needed to let computers understand



The Theological Objection

- The Theological Objection: according to Turing, only humans were given a soul by God. No animal or machine has a soul and that is the reason why they can not think.

The Mathematical Objection

- There are limitations to the powers of any particular machine, even with infinite capacity
- Turing’s Approach: man have limitations and make mistakes too. Maybe in the future there will be machines intelligent enough to compete with humans.

Arguments from Various Disabilities

- Machines can not act out of emotional reasons
- When they act they can not feel
- There are no emotional consequences
- Turing's Approach: we can not know how a machine feels since we are not machines. Machines are limited because of 'the very small capacity of most machines'

Lady Lovelace’s Objection 1

- “Computers can’t be creative. For to be creative is to *originate* something. But computers originate nothing. They merely follow the programs given to them.”

- Turing’s approach: if we could add the possibility to learn and reason to a machine, it could learn everything from scratch like a newborn child

Lady Lovelace's Objection 2

- "machines can never 'take us by surprise'"
- Turing's approach: computers could still surprise humans, in particular where the consequences of different facts are not immediately recognizable.

Continuity with the Nervous System

- The nervous system is certainly not a discrete-state machine
- Turing’s approach: this fact will not make a difference in the imitation game

Conclusions

- Humans can't hyper compute, because
  - They can't really generate truly random numbers
  - They can't really reason about infinity
  - They can't solve the halting problem
- Maybe computers can think in the future, but I'm quite pessimistic about it

The End