

What Are Computers, Really?

Clarissa Littler

6-23-2015

Fundamental Questions

- What are the limits of computers?

Fundamental Questions

- What are the limits of computers?
- Are there problems a computer can never solve?

Fundamental Questions

- What are the limits of computers?
- Are there problems a computer can never solve?
- Do all programming languages describe the same thing?

Fundamental Questions

- What are the limits of computers?
- Are there problems a computer can never solve?
- Do all programming languages describe the same thing?
- What even **are** programs?

How We'll Answer Them

- Give intuitive criterion for “computability” as finite process

How We'll Answer Them

- Give intuitive criterion for “computability” as finite process
- Build up a definition of computation independent of computers

How We'll Answer Them

- Give intuitive criterion for “computability” as finite process
- Build up a definition of computation independent of computers
- Sketch out mathematical models of computation

How We'll Answer Them

- Give intuitive criterion for “computability” as finite process
- Build up a definition of computation independent of computers
- Sketch out mathematical models of computation
- Give examples of non-computable problems

How We'll Answer Them

- Give intuitive criterion for “computability” as finite process
- Build up a definition of computation independent of computers
- Sketch out mathematical models of computation
- Give examples of non-computable problems
- Discuss the implications and limits of our knowledge of computability

Computation Sounds Like Computer

- Computation is what computers do

Computation Sounds Like Computer

- Computation is what computers do
- A program *describes* a computation

Computation Sounds Like Computer

- Computation is what computers do
- A program *describes* a computation
- The limits of computation are limits of **description**

Computation Sounds Like Computer

- Computation is what computers do
- A program *describes* a computation
- The limits of computation are limits of **description**
- What processes can be **described** in a finite way with a finite **implementation**?

Finite Processes

- Why finite?

Finite Processes

- Why finite?
- Because we can't write an infinite amount of text!

Finite Processes

- Why finite?
- Because we can't write an infinite amount of text!
- We're restricted to

Finite Processes

- Why finite?
- Because we can't write an infinite amount of text!
- We're restricted to
 - finite **alphabet**

Finite Processes

- Why finite?
- Because we can't write an infinite amount of text!
- We're restricted to
 - finite **alphabet**
 - finite **length**

Examples of Finite Processes

Recipe as Finite Process

Cook celery and onion together til soft, then add frozen spinach and cook to get some of the moisture out and reduce volume add broth lentils cilantro and other spices, stir thoroughly, throw bay leaves on top.

Cook for 40 minutes

Turn off heat, wait til it stops bubbling and blend thoroughly.

Cook for 5-10 minutes after blending

Examples of Finite Processes

Another finite process

Walking directions

via SW 6th Ave and SW Clay St 12 min
0.5 mile

Show terrain

Use caution - may involve errors or sections not suited for walking

Pioneer Courthouse Square
701 Southwest 6th Avenue, Portland, OR 97204

Head south on SW 6th Ave toward SW Yamhill St
0.4 mi

Turn left onto SW Clay St
0.2 mi

Turn right onto SW 3rd Ave
92 ft

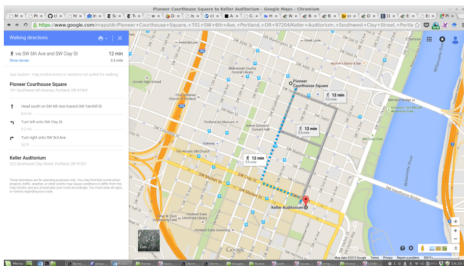
Keller Auditorium
222 Southwest Clay Street, Portland, OR 97201

These directions are for planning purposes only. You may find that construction, projects, traffic, weather, or other events may cause conditions to differ from the map results, and you should plan your route accordingly. You must obey all signs or notices regarding your route.

Examples of Finite Processes

Examples of Finite Processes

Another finite process

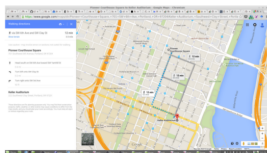


Examples of Finite Processes

Examples of Finite Processes

Examples of Finite Processes

Another finite process



Settings

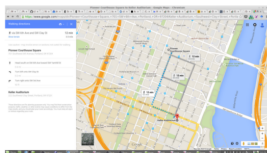


Examples of Finite Processes

Examples of Finite Processes

Examples of Finite Processes

Another finite process



Settings

I'm done with this joke now, I promise.

Examples of Finite Processes

$$\begin{aligned} 5 + 10 &= 4 + 11 \\ &= 3 + 12 \\ &= 2 + 13 \\ &= 1 + 14 \\ &= 0 + 15 \\ &= 15 \end{aligned}$$

Examples of Finite Processes

The following Haskell snippet that evaluates the sum of the integers from 1 to 10 is **also** a finite process

```
let f x = sum [1..x] in f 10  
55
```

Examples of Finite Processes

Many more examples exist in the wild including:

- counting on your fingers

Examples of Finite Processes

Many more examples exist in the wild including:

- counting on your fingers
- long division

Examples of Finite Processes

Many more examples exist in the wild including:

- counting on your fingers
- long division
- sorting your vinyl collection with a bucket sort

Examples of Finite Processes

Many more examples exist in the wild including:

- counting on your fingers
- long division
- sorting your vinyl collection with a bucket sort
- compiling code

Examples of Finite Processes

Many more examples exist in the wild including:

- counting on your fingers
- long division
- sorting your vinyl collection with a bucket sort
- compiling code

Qualities of Finite Processes

- Informal criterion for a “finite process”
 - Finite Time
 - Finite Resources
 - Finite Directions

What Does Finite Mean?

- A quantity is finite when it is “measurable”
 - Counting
 - Weighing
 - Timing

Finite Time

- Finite process produces **output** in finite time
- Finite time:
 - Counting on your fingers
 - Sorting vinyl
 - Compiling
 - Walking to a friend's house

Why Finite Time?

Only actions taking finite time can actually be finished because that's how our universe works.

Finite Resources

- Finite processes only use finite resources
 - scratch paper
 - materials
 - RAM
 - disk space

Why Finite Resources

No computer and no physical process that we know of can use an infinite quantity, thus infinite resources shouldn't be allowed in computation.

Finite Directions

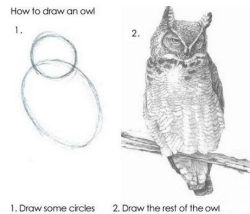
- Finite processes are only allowed to have a finite number of steps in their description
 - Directions
 - Recipes
 - **Programs**
 - note: a `while` loop is a finite!

Why Finite Directions?

Any process that has an infinite number of steps in its directions must violate either the finite time constraint, finite resources, or both.

What Next?

- These are rules of thumb
- But how do we **actually** specify a process?
- Most directions too broad:
 - driving directions
 - cooking directions
- Need instructions simple enough for a machine



Why a Model?

- We need a precise, mathematical formulation of descriptions
 - Need to know if all of our “pieces” are finitary
 -

Why a Model?

- We need a precise, mathematical formulation of descriptions
 - Need to know if all of our “pieces” are finitary
 -

What Is a Model?

The Need for a Model of Computation

- Are C, Java, Scheme, and Erlang all describing the same programs?
- Need an independent, rigorous description of computation
- Partial recursive functions
- Lambda calculus
- Turing's automatic machines

Turing and His Automatic Machines

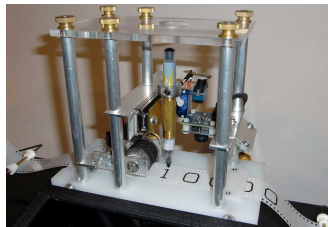
- Turing's 1936 paper "On Computable Numbers, with an Application to the [Decision Problem]" [3]
- Turing's "automatic machines" were addressing problems in foundational logic
 - The "decision problem"
- Automatic machines weren't **actually** stand-ins for modern computers
- Turing was inspired by human computers

Human Computers

Decisions and Problems

A Turing Machine

- Arbitrary amount of tape
- Reads and writes from only once cell at a time
- Only has a finite “alphabet” of symbols
- Has a finite number of states for deciding next move



Historic Importance of Turing Machines

Human Computers and Finite Processes

Specifications of Problems

- Specification **precisely** describes the problem
- Implementation is the solution to the problem
- Fundamental question of computing:
 - what specifications have a computable implementation?

Specifications vs. Implementations

Mathematical addition is a specification, how you perform the addition is an implementation

- counting on fingers
- repeatedly adding 1
-

Computations Solve Specifications

The Halting Problem

No Perfect Virus Scanner

Rice's Theorem

The Church-Turing Thesis

Original Formulation

There is no model of computation more expressive than Turing machines (equivalently, the lambda calculus). [2]

Equivalent Formulation

Equivalent formulation: no programming language can be more powerful than a Turing machine

Church-Turing as Physics

Strong Church-Turing Thesis

The laws of physics are inherently computable and there is no physical process that cannot be computed by some algorithm.

- is this actually true?

Church-Turing as Physics

Strong Church-Turing Thesis

The laws of physics are inherently computable and there is no physical process that cannot be computed by some algorithm.

- is this actually true?
- we literally have no idea!

Church-Turing as Physics

Strong Church-Turing Thesis

The laws of physics are inherently computable and there is no physical process that cannot be computed by some algorithm.

- is this actually true?
- we literally have no idea!
- strong Church-Turing thesis has major implications for physics

Church-Turing as Physics

Strong Church-Turing Thesis

The laws of physics are inherently computable and there is no physical process that cannot be computed by some algorithm.

- is this actually true?
- we literally have no idea!
- strong Church-Turing thesis has major implications for physics
 - reality must be “discrete”

Church-Turing as Physics

Strong Church-Turing Thesis

The laws of physics are inherently computable and there is no physical process that cannot be computed by some algorithm.

- is this actually true?
- we literally have no idea!
- strong Church-Turing thesis has major implications for physics
 - reality must be “discrete”
 - real numbers are approximations at scale

Church-Turing as Cognition

- Are brains computable?

Church-Turing as Cognition

- Are brains computable?
- Currently an unknown question

Church-Turing as Cognition

- Are brains computable?
- Currently an unknown question
- Does free will actually exist or is it an illusion?

Is Strong AI Possible?

- Can we make a machine intelligence comparable to our own?

Is Strong AI Possible?

- Can we make a machine intelligence comparable to our own?
- Is human intelligence computable?

Is Strong AI Possible?

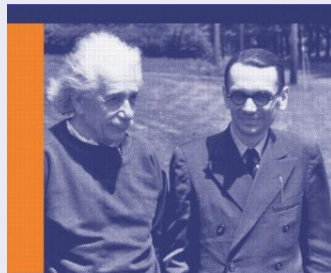
- Can we make a machine intelligence comparable to our own?
- Is human intelligence computable?
- Science fiction assumes so!

Is Strong AI Possible?

- Can we make a machine intelligence comparable to our own?
- Is human intelligence computable?
- Science fiction assumes so!
- Deep philosophic and physical implications

Skirting Computability

Goedel's Way



Gödel's Way

Exploits into an
undecidable world

Gregory Chaitin
Newton da Costa
Francisco Antonio Doria

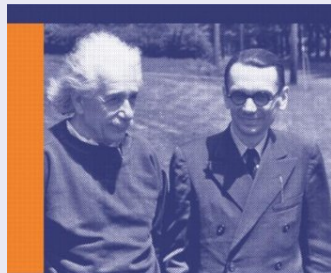


Hyper-Turing Computation

- Is computation (in the Church-Turing sense) complete?

Skirting Computability

Goedel's Way



Gödel's Way

Exploits into an
undecidable world

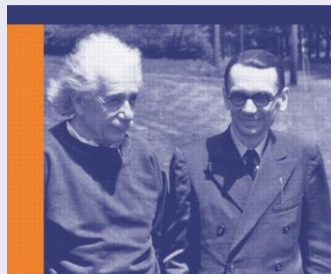
Gregory Chaitin
Newton da Costa
Francisco Antonio Doria



Hyper-Turing Computation

- Is computation (in the Church-Turing sense) complete?
- Is physics computable?

Goedel's Way



Gödel's Way

Exploits into an
undecidable world

Gregory Chaitin
Newton da Costa
Francisco Antonio Doria

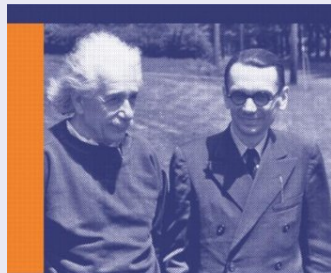


Hyper-Turing Computation

- Is computation (in the Church-Turing sense) complete?
- Is physics computable?
 - If yes, then no

Skirting Computability

Goedel's Way



Gödel's Way

Exploits into an
undecidable world

Gregory Chaitin
Newton da Costa
Francisco Antonio Doria

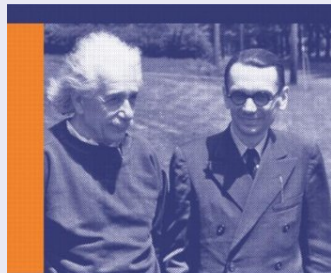


Hyper-Turing Computation

- Is computation (in the Church-Turing sense) complete?
- Is physics computable?
 - If yes, then no
 - If no, then **maybe** [1]

Skirting Computability

Goedel's Way



Gödel's Way

Exploits into an
undecidable world

Gregory Chaitin
Newton da Costa
Francisco Antonio Doria



Hyper-Turing Computation

- Is computation (in the Church-Turing sense) complete?
- Is physics computable?
 - If yes, then no
 - If no, then **maybe** [1]
- Hyper-Turing computation

In Conclusion

- Computation is a thing that exists outside computers

In Conclusion

- Computation is a thing that exists outside computers
- The mathematics of finite methods

In Conclusion

- Computation is a thing that exists outside computers
- The mathematics of finite methods
- Computation has **limits**

In Conclusion

- Computation is a thing that exists outside computers
- The mathematics of finite methods
- Computation has **limits**
- The limits of computation are understood

In Conclusion

- Computation is a thing that exists outside computers
- The mathematics of finite methods
- Computation has **limits**
- The limits of computation are understood
- How computation relates to the laws of the universe?

In Conclusion

- Computation is a thing that exists outside computers
- The mathematics of finite methods
- Computation has **limits**
- The limits of computation are understood
- How computation relates to the laws of the universe?
 - Much more unknown

Any Questions?

Bibliography



Gregory Chaitin, Francisco A Doria, and Newton CA da Costa.
Gödel's way: exploits into an undecidable world.
CRC Press, 2011.



Stephen Cole Kleene, NG de Bruijn, J de Groot, and
Adriaan Cornelis Zaanen.
Introduction to metamathematics.
1952.



Alan Mathison Turing.
On computable numbers, with an application to the
entscheidungsproblem.
J. of Math, 58(345-363):5, 1936.