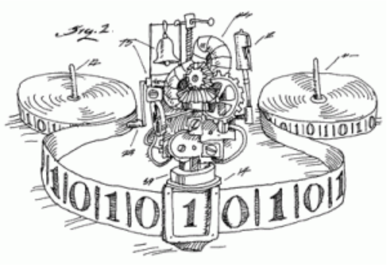


Computability and Busy Beavers

Jean Cardinal
Smart Monday, May 2025

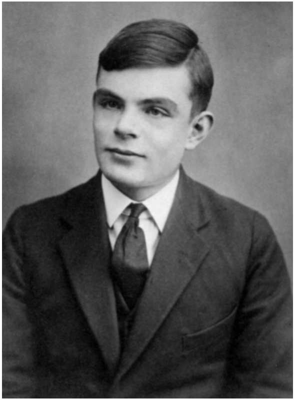
Computability

- Can we compute everything?

Computability

- Can we compute everything?
- What does "computing" mean?

Alan Turing



"On Computable Numbers, with an Application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, Series 2, 42 (1936–7), pp 230–265.

Turing Machines

A **Turing Machine** is composed of:

- A finite *set of states* Q, with three special *start*, *reject*, and *accept* states.
- An infinite *tape*, divided into *cells*, each containing a symbol, either **0** or **1**.
- A *head* is positioned on one of the cells, and can only move left or right.
- A *transition function* from $Q \times \{0, 1\}$ to $Q \times \{0, 1\} \times \{\text{left}, \text{right}\}$.

Turing Machines

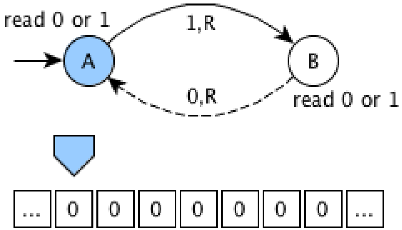
A **Turing Machine** is composed of:

- A finite *set of states* Q, with three special *start*, *reject*, and *accept* states.
- An infinite *tape*, divided into *cells*, each containing a symbol, either **0** or **1**.
- A *head* is positioned on one of the cells, and can only move left or right.
- A *transition function* from $Q \times \{0, 1\}$ to $Q \times \{0, 1\} \times \{\text{left}, \text{right}\}$.

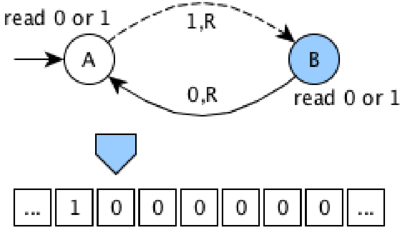
At every time step, the combination of the current state and the tape symbol pointed by the head define the operations to perform:

- change to a new state in Q,
- replace the symbol on the tape by a new symbol,
- move the head of one position left or right.

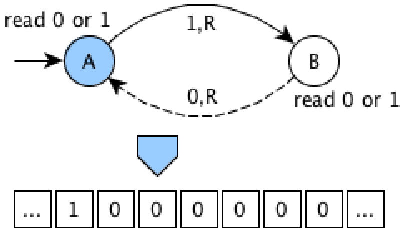
Example



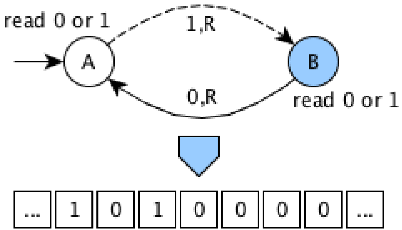
Example



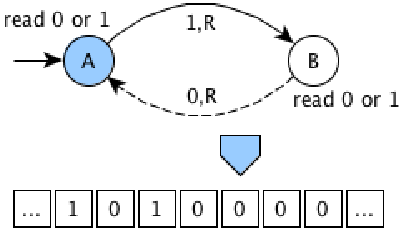
Example



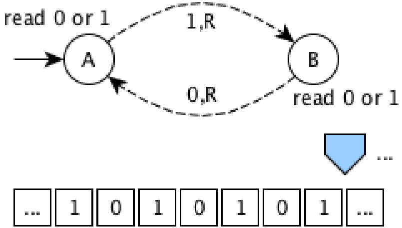
Example



Example



Example



A Three-state Turing Machine

Tape symbol	**Current state A**			**Current state B**			**Current state C**		
	Write	Move	Next	Write	Move	Next	Write	Move	Next
0	1	R	B	1	L	A	1	L	B
1	1	L	C	1	R	B	1	R	HALT

Decidability

- **Church-Turing thesis:** A function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine.

Decidability

- **Church-Turing thesis:** A function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine.
- Some functions are not computable!

Decidability

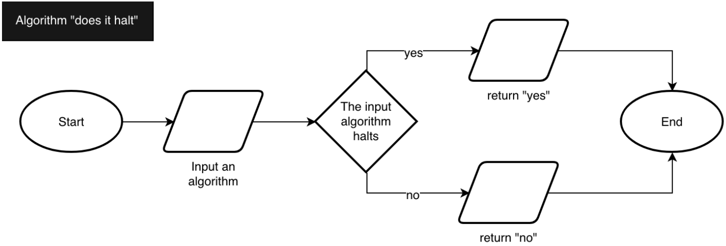
- **Church-Turing thesis:** A function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine.
- Some functions are not computable!
- Related to completeness in axiomatic systems (cf. Gödel's Theorem).

Halting problem

- Given a Turing Machine (TM), does it ever halt?

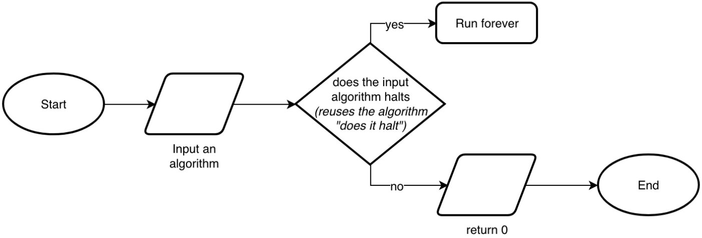
Halting problem

- Given a Turing Machine (TM), does it ever halt?
- Suppose there exists a TM that decides whether any given TM halts.



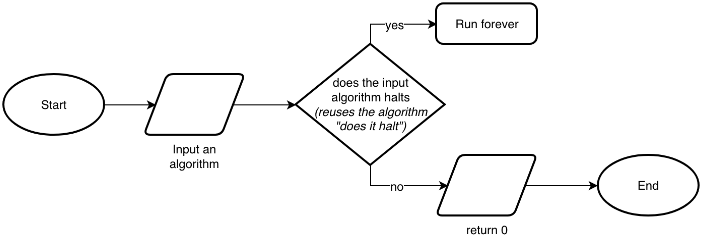
Halting problem

- Construct another TM that does not halt if the answer is yes.



Halting problem

- Construct another TM that does not halt if the answer is yes.



- Give this TM as input to itself. It halts if and only if it does not!

Busy Beaver

- $\Sigma(n)$ is the *Maximum number of 1s written by an n-state TM that halts*.

Busy Beaver

- $\Sigma(n)$ is the *Maximum number of 1s written by an n-state TM that halts*.
- $\text{BB}(n)$ is the *Maximum number of steps taken by an n-state TM that halts*.

Busy Beaver

- $\Sigma(n)$ is the *Maximum number of 1s written by an n-state TM that halts*.
- $BB(n)$ is the *Maximum number of steps taken by an n-state TM that halts*.
- The previous example gives the value of $\Sigma(3)$!

Small Values

n	1	2	3	4	5
$\Sigma(n)$	1	4	6	13	$>$4,098
$\mathrm{BB}(n)$	1	6	21	107	47,176,870

Growth

It is known that

$$\mathrm{BB}(6) > \underbrace{10^{10^{10^{\cdot^{\cdot^{\cdot^{10}}}}}}}_{\text{15 times}}$$

(Kropitz 2022)

Growth

It is known that

$$\mathrm{BB}(6) > \underbrace{10^{10^{10^{\cdot^{\cdot^{\cdot^{10}}}}}}}_{\text{15 times}}$$

(Kropitz 2022)

- Given oracle access to any function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) > \mathrm{BB}(n)$ for all n, one can solve the halting problem. Hence, no such f can be computable!
- The busy beaver function *grows faster than any computable function*!

Growth

It is known that

$$\mathrm{BB}(6) > \underbrace{10^{10^{10^{\cdot^{\cdot^{\cdot^{10}}}}}}}_{15 \text{ times}}$$

(Kropitz 2022)

- Given oracle access to any function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) > \mathrm{BB}(n)$ for all n, one can solve the halting problem. Hence, no such f can be computable!
- The busy beaver function *grows faster than any computable function*!
- More precisely: for any computable function f, there exists n_0 such that $\mathrm{BB}(n) > f(n)$ for all $n > n_0$.

Growth

It is known that

$$\mathrm{BB}(6) > \underbrace{10^{10^{10^{\cdot^{\cdot^{\cdot^{10}}}}}}}_{\text{15 times}}$$

(Kropitz 2022)

- Given oracle access to any function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) > \mathrm{BB}(n)$ for all n, one can solve the halting problem. Hence, no such f can be computable!
- The busy beaver function *grows faster than any computable function*!
- More precisely: for any computable function f, there exists n_0 such that $\mathrm{BB}(n) > f(n)$ for all $n > n_0$.

BB(5)

- The value of $\mathrm{BB}(5) = 47,176,870$ has been found in... 2024.

BB(5)

- The value of $\mathrm{BB}(5) = 47,176,870$ has been found in... 2024.
- A massive international collaboration set up by french researcher Tristan Stérin, involving heavy computational and formal machinery.

BB(5)

- The value of $BB(5) = 47,176,870$ has been found in... 2024.
- A massive international collaboration set up by french researcher Tristan Stérin, involving heavy computational and formal machinery.

Aaronson:

> *"It's a fair question whether humans will ever know the value of BB(6)."*

BB(5)

- The value of $\mathrm{BB}(5) = 47{,}176{,}870$ has been found in... 2024.
- A massive international collaboration set up by french researcher Tristan Stérin, involving heavy computational and formal machinery.

Aaronson:

> *"It's a fair question whether humans will ever know the value of $BB(6)$."*
>
> *"If and when artificial superintelligences take over the world, they can worry about the value of $BB(6)$. And then God can worry about the value of $BB(7)$."*

Encoding Consistency of Mathematics

- There exists a value n such that $\mathrm{BB}(n)$ is uncomputable.

Encoding Consistency of Mathematics

- There exists a value n such that BB(n) is uncomputable.
- There exists a TM with some number n of states that enumerates all possible theorems in an axiomatic system (here ZF), and stops if a contradiction is met.
- Determining the value of BB(n) would then define a procedure for determining the consistency of the system, which – assuming it is consistent – is known to be impossible.

A Concrete Bound

- In 2016, Yedidia and Aaronson gave the first such explicit value: $n = 7910$.

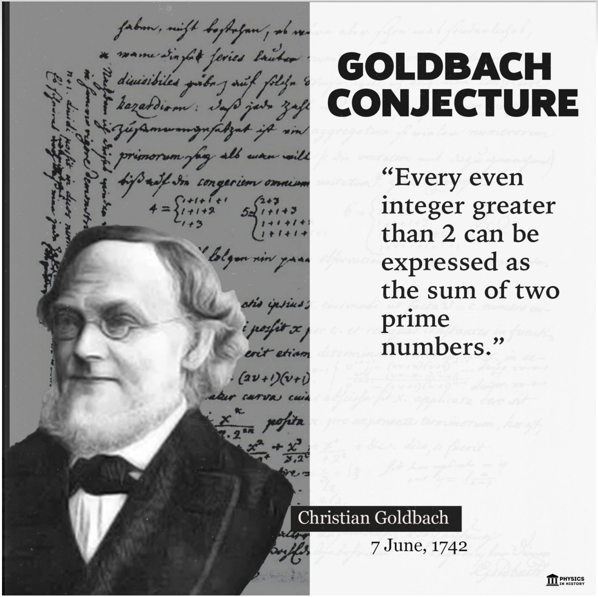
A Concrete Bound

- In 2016, Yedidia and Aaronson gave the first such explicit value: $n = 7910$.
- It was then improved to 1919, 748, and 745. This latter record was achieved by Johannes Riebel, in a bachelor thesis at the University of Augsburg.

A Concrete Bound

- In 2016, Yedidia and Aaronson gave the first such explicit value: $n = 7910$.
- It was then improved to 1919, 748, and 745. This latter record was achieved by Johannes Riebel, in a bachelor thesis at the University of Augsburg.

In short: There is an explicit 745-state Turing machine that halts if ZF is inconsistent. Thus, assuming ZF is consistent, ZF cannot prove the value of $BB(745)$.



(Physics in history)

Goldbach Conjecture

Suppose we have an n-state TM that checks all possible natural numbers one by one, and stops whenever it finds a counterexample. Then after $\mathrm{BB}(n)$ steps, we know for sure the conjecture holds! Such a machine has been designed with $n = 25$.

Goldbach Conjecture

Aaronson, May 4th, 2016:

> *I guesstimated that there should be a simple 100-state machine for Goldbach's Conjecture. But ... how to say this?*

Goldbach Conjecture

Aaronson, May 4th, 2016:

> *I guesstimated that there should be a simple 100-state machine for Goldbach's Conjecture. But ... how to say this?*

Jared S, May 6th, 2016, 5:47am:

> *I have a 73-state, two-symbol, one-tape TM that halts iff Goldbach's conjecture is false.*

Goldbach Conjecture

Aaronson, May 4th, 2016:

> *I guesstimated that there should be a simple 100-state machine for Goldbach's Conjecture. But ... how to say this?*

Jared S, May 6th, 2016, 5:47am:

> *I have a 73-state, two-symbol, one-tape TM that halts iff Goldbach's conjecture is false.*

code golf addict, May 6th, 2016 7:45am:

> *I've got 31 states for Goldbach.*

Goldbach Conjecture

Aaronson, May 4th, 2016:

> *I guesstimated that there should be a simple 100-state machine for Goldbach's Conjecture. But ... how to say this?*

Jared S, May 6th, 2016, 5:47am:

> *I have a 73-state, two-symbol, one-tape TM that halts iff Goldbach's conjecture is false.*

code golf addict, May 6th, 2016 7:45am:

> *I've got 31 states for Goldbach.*

code golf addict, May 8th, 2016 at 1:46 pm

> *I've got Goldbach down to 27 now.*

Current record: 25 states!

Pointers

Textbook Michael Sipser, *"Introduction to the Theory of Computation, 3rd ed."*

Wikipedia https://en.wikipedia.org/wiki/Busy_beaver

OEIS https://oeis.org/search?q=busy+beaver&go=Search

Aaronson's blog https://scottaaronson.blog/?p=2725 and https://scottaaronson.blog/?p=8088

Riebel's BA thesis https://www.ingo-blechschmidt.eu/assets/bachelor-thesis-undecidability-bb748.pdf

bbchallenge website https://bbchallenge.org/8041034, see also announcement here

25-state Goldbach machine https://github.com/lengyijun/goldbach_tm

Quanta magazine video https://www.youtube.com/watch?v=rmx3FBPzDuk