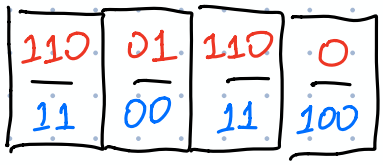
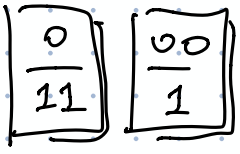
NP-hard ≅ no fast algorithm

Undecidable = no algorithm

Post Correspondence



110011100



NO ALGORITHM

COLLATZ(n):

```
while n>1
    if n is even
        n ← n/2
    else
        n ← 3n+1
return True!
```

7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

FRACTRAN

John Conway

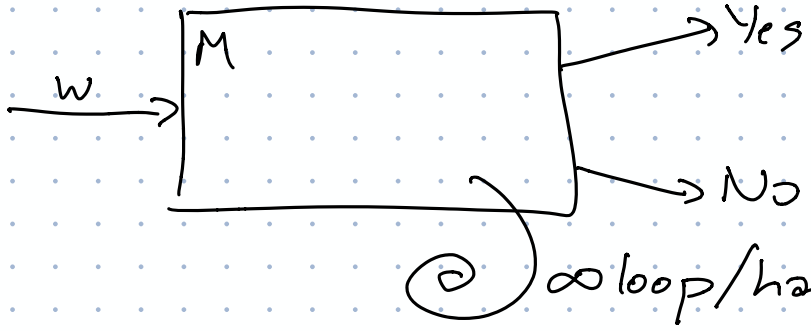
Code = Data

M — machine / executable program / device

⟨M⟩ — encoding / source code / description

Yes/No questions

~~When~~ IF a machine halts, either accepts or rejects.



Halting problem: We want an algorithm that decides whether M halts, given ⟨M⟩ as input.

SELFREJECT: Given encoding ⟨M⟩ of some machine M, decide whether M rejects ⟨M⟩.

Suppose SR decides SELFREJECT

ACCEPT(SR) = SELFREJECT
Diverge(SR) = ∅

For all M:

SR accepts ⟨M⟩ ⇔ M rejects ⟨M⟩

SR rejects ⟨M⟩ ⇔ M does not reject ⟨M⟩

In particular when M = SR:

SR accepts ⟨SR⟩ ⇔ SR rejects ⟨SR⟩

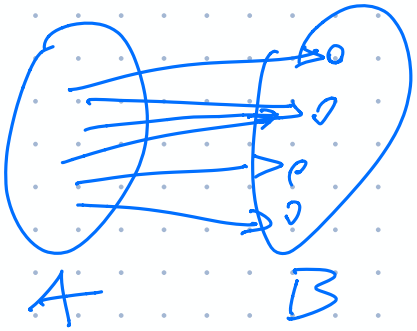
SR rejects ⟨SR⟩ ⇔ SR does not reject ⟨SR⟩

Diagonalization — Cantor

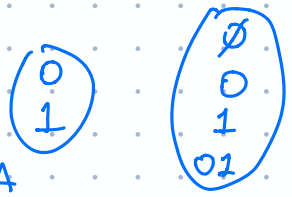
0.11011000...
0.0000100...
0.10100011 0...
0.0000000...
0.11111...

0.0101011...

$F: A \to B$ is a surjection ("onto") if every element of B is in the image



Cantor's Theorem: There is no onto $F: A \to 2^A$ for all sets A



Proof: Spose there is an onto $f: A \to 2^A$

Call $x \in A$ happy if $x \in f(x)$

$x \in A$ sad if $x \notin f(x)$

Let S be set of all sad elements

Onto: Pick $y \in A$ such that $f(y) = S$

$y \in S \Leftrightarrow y$ sad $\Leftrightarrow y \notin f(y)$

forall x: $x \in f(y) \Leftrightarrow x \in S \Leftrightarrow x \notin f(x)$

$x = y$ $y \in f(y) \Leftrightarrow y \in S \Leftrightarrow y \notin f(y)$ Contradiction

Non-president's TCSO doesn't exist!

HALT = Given encoding $\langle M \rangle$ and string w
does M halt on w?

undecidable