Turing's Idea

COGS 200

A String Game

Hofstadter's *MIU:*

- MIU is a game with strings of symbols.
- The symbols are 'M', 'U' and 'I'.
- There are rules that enable you to add to and take from the string you've currently got.
- You start with 'MI'.

Rules of MIU

- If the last letter of your string is I, you can add a U on the end.
- Suppose you have 'M*x*' (where *x* is any string). You may add 'M*xx*' to your collection (where *xx* is that string twice).
- If 'III' occurs in any of the strings in your collection, you may make a new string with 'U' in place of 'III'.
- If 'UU' occurs inside one of your strings, you can drop it.

Three MIU questions

Start with 'MI'

Can you get 'MUIU'?

Can you get 'MUIIIU'?

Rules of MIU

If the last letter of your string is I, you can add a U on the end.

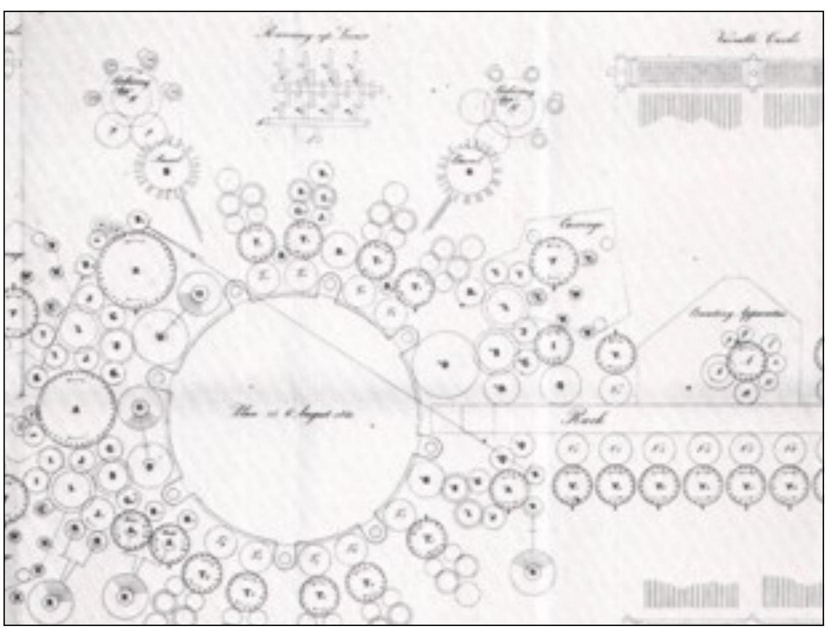
Suppose you have 'M*x*' (where *x* is any string). You may add 'M*xx*' to your collection (where *xx* is that string twice).

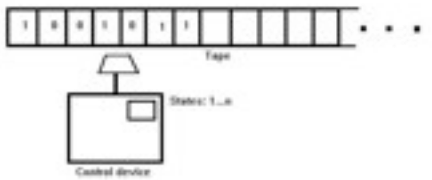
If 'III' occurs in any of the strings in your collection, you may make a new string with 'U' in place of 'III'.

If 'UU' occurs inside one of your strings, you can drop it.

Ada Lovelace

1815-1852





Turing Machines

Never mind, for now, about running out of paper.

The Turing Machine has an infinite tape, divided up into squares.

But it can only see one square at a time.

And all it can see is whether the square is empty, or has been marked.

The Head of the Turing Machine

Can write on a square (if that square is empty).

Can erase what's written in a square (if that square is marked).

Can move up and down the tape (one square at a time).

Can halt.

Which of these it will do depends on what it sees on the one square that it can survey.

The States of a Turing Machine

Specify one action (read, write, forward, back or halt) if you see a 0 ...

... and one action (from the same repertoire) if you see a 1.

One State Machines

- Moves the head to the end of the first block of 1s.

0	1
Halt	Forward

One State Machines

- Never halts!

0	1
Forward	Forward

Adding More States

- Performs addition!

	0	1
S1	Forward (S2)	Erase (S1)
S2	Write (S3)	Forward (S2)
S3	Halt	Back (S3)

Exercise

Write the table for a Turing machine that, if started on the leftmost square of a block of *n* dashes, will halt only if the number of dashes on its tape is odd.

Assume that the rest of the tape is blank.

How would you do doubling?

Turing's Thesis

Every recursively definable function is Turing computable.

"The astonishing thing about computers is that any information that can be stated in a language can be encoded in such a system, and any information-processing task that can be solved by explicit rules can be programmed" (Searle, 1990)

Two Remarkable Facts

It is possible to encode the table, and put it on the tape.

There is a table that, when run on a tape with an encoded table, and then an input, the machine will imitate whatever machine operates with that table.

That is, there is a **Universal Turing Machine**.

And there are **uncomputable functions**.

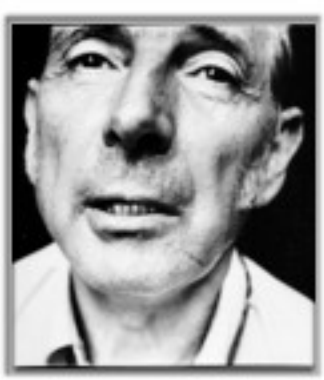
Valiant

"Many of the foremost thinkers of the early part of the twentieth century had wondered, somewhat informally, whether mechanical procedures existed for resolving all mathematically well-posed questions. Some such as the philosopher Bertrand Russell and the mathematician David Hilbert, were optimistic. Turing's discovery that one could define precisely what such an assertion meant, and then *prove* that such a statement was false, has revolutionary implications. The shock of this is still taking its time to permeate the community of the educated."

Valiant 2013, p. 25

The Chinese Room

John Searle



“I have not tried to prove that “a computer cannot think.” Since anything that can be simulated computationally can be described as a computer, and since our brains can at some levels be simulated, it follows trivially that our brains are computers, and they can certainly think. But from the fact that a system can be simulated by symbol manipulation, and the fact that it is thinking, it does not follow that thinking is equivalent to formal symbol manipulation.”

(Searle, p. 27)

Hofstadster’s ‘*pq-’*

Again, just three symbols:

‘ ♭ ♯ - ’

If ***x*** is any number of hyphens, you may start with:

‘***x*** ♭ - ♯ ***x***-’

If ***x***, ***y*** and ***z*** are strings of hyphens, then:

if you have ‘***x*** ♭ ***y*** ♯ ***z***’

you may write: ‘***x*** ♭ ***y*** - ♯ ***z***-’.

Two pq- problems

Can you get: '**--- ♭ --- ♯ -----**'?

Is there a general rule that determines which strings you can get?

Hofstadster's '*pq-*'

Again, just three symbols:

'♭ ♯ **-**'

If ***x*** is any number of hyphens, you may start with:

'***x*** ♭ **-** ♯ ***x*-**'

If ***x***, ***y*** and ***z*** are strings of hyphens, then:

if you have '***x*** ♭ ***y*** ♯ ***z***'

you may write: '***x*** ♭ ***y* -** ♯ ***z*-**'.

The significance of 'pq-'

Interpret ' ♭ ' as '+' and ' ♯ ' as '='.

The strings in this game are true statements about addition.

(And all the true statements that can be framed in this impoverished language can be derived.)

BUT - noticing this was an intellectual achievement, distinct from competent application of the rules of the game.