From Mind to Turing to Mind

Henk Barendregt

Faculty of Science
Radboud University
Nijmegen, The Netherlands

Many *numerical problems* can be answered by computing

"What is the area of a circle with radius $4\,\mathrm{yd}$?"

Answer: $4^2\pi\,\mathrm{yd}^2 = 50.2654824\,\mathrm{yd}^2$

Also many *qualitative problems* may be answered by computing

"Are points $A = (x_1, y_1)$, $B = (x_2, y_2)$ and $C = (x_3, y_3)$ in $\mathbb{R}^2$ collinear

i.e. do they lie on a straight line?"

Answer: if and only if $(x_1 - x_3)(y_2 - y_3) = (x_2 - x_3)(y_1 - y_3)$

Also for *daily life actions* or in the *performing arts* and *sports* one needs computations

Musicians can train themselves to be accurate in the sub-millisecond range

Hence one needs fast and accurate computations

Bartok: Sonata, Ivry Gitlis violin

Ravel: Ondine from Gaspard de la Nuit, John Kane piano

Leibniz (1646-1716) conjectured:

All properly stated problems can be answered by computing (calculemus!)

He wanted to construct:

a universal language L for stating problems precisely

a machine M answering all these problems by computing

The first question Leibniz wanted to ask to such a machine is said to be "Does God exist?"
Quite daring around 1700 to ask this question to a machine!

Science Fiction *Answer*, F. Brown [1954]: "Now there is a God!"

If we restrict ourselves to mathematical problems, then there is such an L

If we restrict ourselves to *numerical problems* then there is such an M

A computer with a software package like Mathematica or Maple

However, Turing [1936]: for *qualitative mathematical problems* M is impossible

Leibniz (1646-1716) conjectured:

All properly stated problems can be answered by computing (calculemus!)

He wanted to construct:

a universal language L for stating problems precisely

a machine M answering all these problems by computing

If we restrict ourselves to mathematical problems, then there is such an L

If we restrict ourselves to *numerical problems* then there is such an M

A computer with a software package like Mathematica or Maple

However, Turing [1936]: for *qualitative mathematical problems* M is impossible

E.g. "Are there twin primes (differing by 2 like $\{11, 13\}$) greater than googolplex $10^{10^{100}}$?"

is an example of such a qualitative problem

Turing [1936]: For *qualitative mathematical problems* M is impossible

Turing's negative answer about the qualitative problems

made possible the positive answer for the numerical problems

There is also a positive answer for a subclass of the qualitative problems

(symbolic computing, e.g. "Are points A, B, C collinear?")

So we have

numerical, symbolic, and qualitative mathematical problems

the latter involving quantifiers "for all" ($\forall$), "there exists" ($\exists$)

These have to do with mathematical infinity

Still we can answer some problems about infinity, by proving

Euclid: There are infinitely many primes: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, . . .

Hilbert believed that eventually everything can be known (by proving or computing)

"Instead of the sad 'We don't know', my attitude is: 'We *have* to know, we *will* know'!"

How did Turing prove his negative result?

How could it have so much positive spin off?

Turing did the following

- ! Gave a well-motivated analysis of computability via *Turing Machines* (TM) an idealized class of machines (with infinite memory)
- !! Constructed a *universal Turing Machine* UTM that can simulate any TM via software (*programs*)
- ! Formulated the *halting problem* (HP) that cannot be decided by any TM argument like liar paradox
- ! Concluded qualitative mathematical problems cannot be decided by any TM as the HP is one of them

Therefore Leibniz's ideal cannot be fulfilled for mathematical problems
only be approximated, leave alone for philosophical problems

A *process* reacts to input with actions (in principle for always)

Think of

an animal that looks for food, a mate and tries to avoid being eaten

a machine that controls a non-stop factory

the operating system of a computer

A *computation* can be seen as a terminating process

Think of

an animal that has to decide whether to go into fight or flight

computing $2^{127} - 1 = 170141183460469231731687303715884105727$ (a prime)

Both

computations (as terminating processes) and

proper processes (non-terminating ones)

are interesting

www.youtube.com/watch?v=E3keLeMwfHY

Based on introspection: computing is done in *discrete* steps

A computational process goes from input to output (action)

Let M be a TM. It has

a finite set I of input

a finite set A of actions

The machine M transforms an input (i in I) into an action (a in A)

$$i \xrightarrow{M} a$$

As the same input may give rise at different times to different actions

Turing introduced a set S of *states* indicating how to react to an input

$$(i, s) \xrightarrow{M} (a, s')$$

The machine, given input and state, may choose action and (a new) state

M is given as a finite table of transitions $(i, s; a, s')$

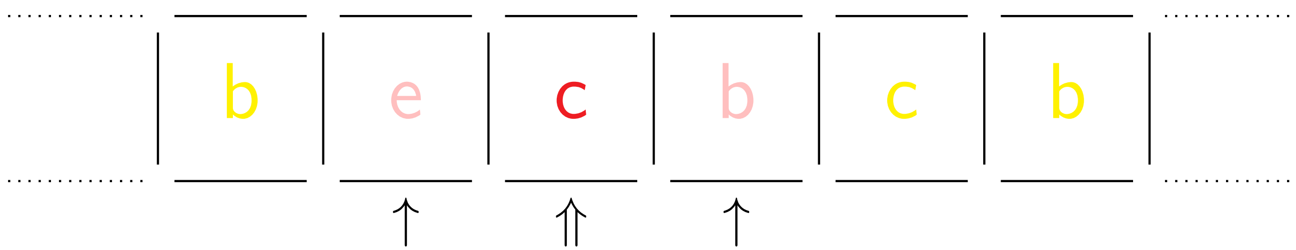
M and has a two-sided infinite memory tape

with cells containing nothing (a blank) or a symbol $i \in I$

and also has a Read/Write device (‘head’) placed on one of the cells

An (*instantaneous*) *configuration* (at a given moment)

is the information on the tape & the position of the head

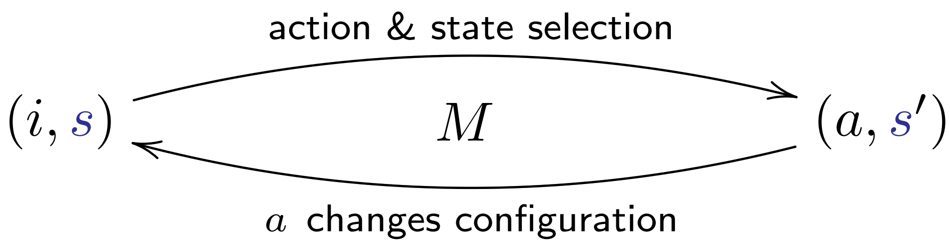


⇑		position of read/write head	Actions	
↑		potential next position of head	L	⇑ goes left
red	letter	focus of attention	R	⇑ goes right
pink	letters	pre-focus of attention	W(c’)	overwrite c by c’
yellow	letters	out of attention		

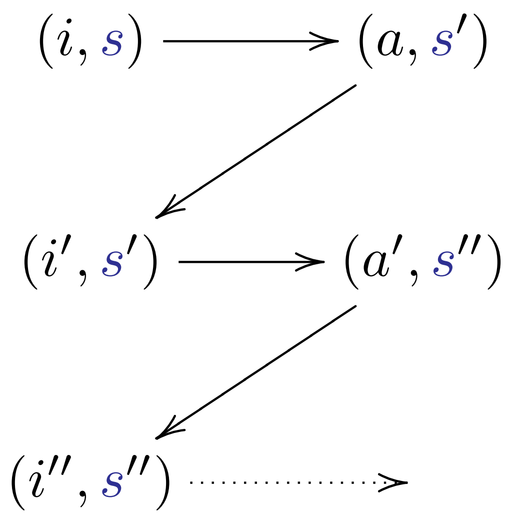
(terminology suggestive for sequel)

Actions modify configuration

The two phases of M combine as follows



giving a ‘scenario’ as follows: c initial configuration, s initial state, f ‘focus of’

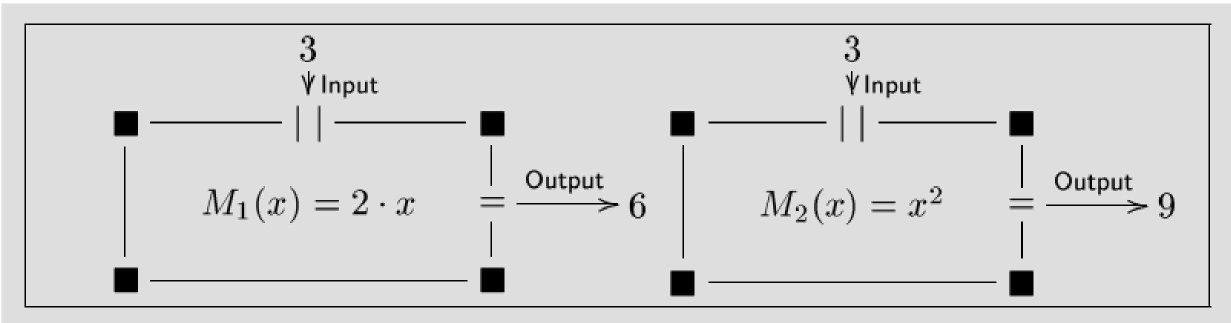


$$i = f(c) \qquad (a, s') = M(i, s)$$

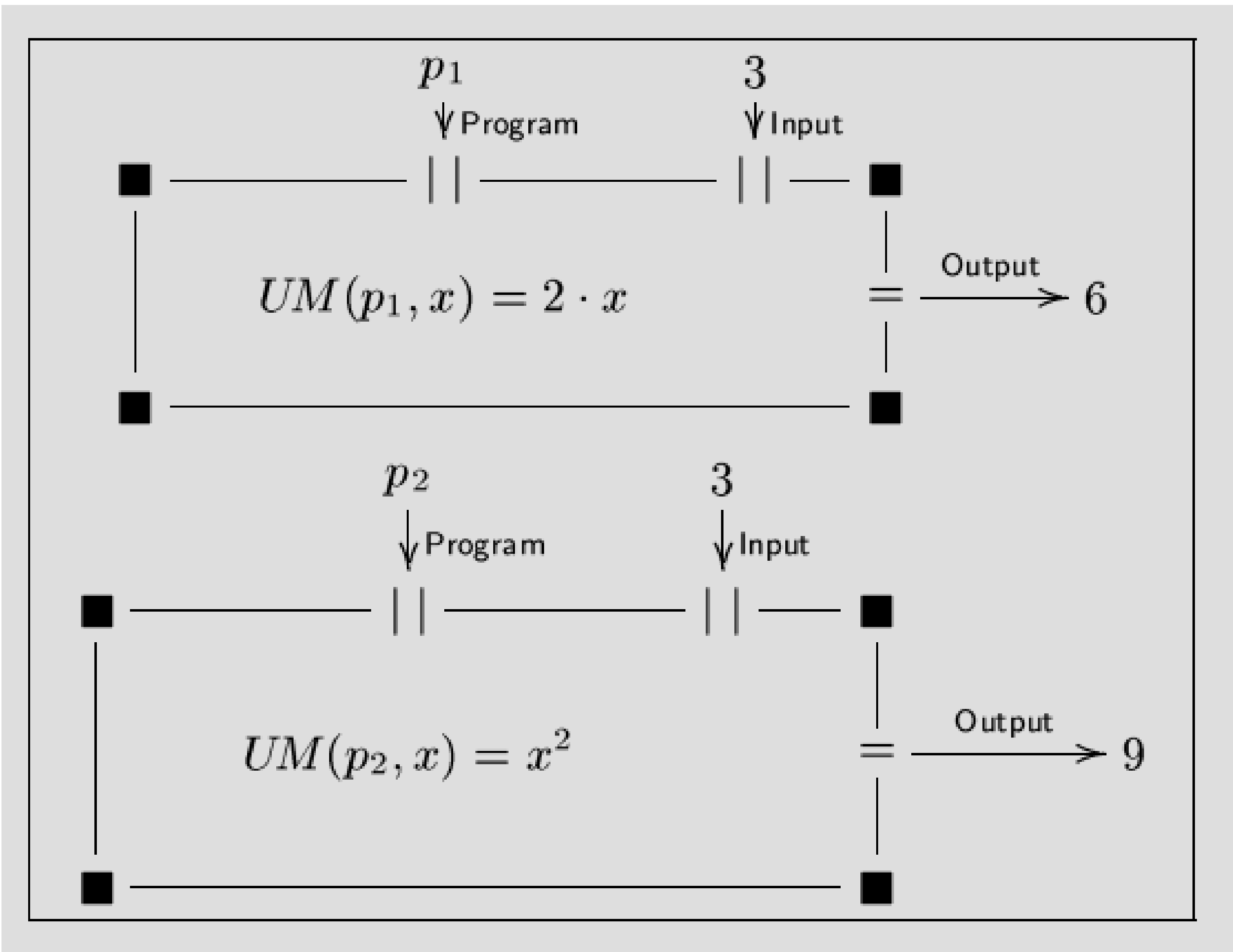
$$c' = a(c),\ i' = f(c') \qquad (a', s'') = M(i', s')$$

$$c'' = a'(c'),\ i'' = f(c'')$$

M *halts* if no more transition is possible. Then *input*: initial c; *output*: final c



Two ad hoc machines M_1, M_2



UM simulating M_1, M_2

Write $M(n)\!\downarrow$ (resp. ($M(n)\!\uparrow$) if machine M with input n does (doesn't) halt

Theorem. The problem $UM(p,n)\!\downarrow$ is non-computable (*undecidable*)

Proof. Suppose $UM(p,n)\!\downarrow$ is computable: then there is a machine H such that

$$\begin{aligned} UM(p,n)\!\downarrow &\iff H(p,n)=1 \\ UM(p,n)\!\uparrow &\iff H(p,n)=0 \end{aligned}$$

Modify H into H' making $H'(p,n)$ run forever if $H(p,n)=1$. Then

$$\begin{aligned} UM(p,n)\!\downarrow &\iff H'(p,n)\!\uparrow \\ UM(p,n)\!\uparrow &\iff H'(p,n)\!\downarrow \end{aligned}$$

Now define $D(n)=H'(n,n)$. It has a program p_D as UM is universal:

$$D(n)=UM(p_D,n) \text{ for all } n$$

We get a contradiction

$$D(p_D)\!\downarrow \iff UM(p_D,p_D)\!\downarrow \iff H'(p_D,p_D)\!\uparrow \iff D(p_D)\!\uparrow \blacksquare$$

$UM(p,n)\downarrow \iff$ there exists a number k such that after k cycles $UM(p,n)$ halts

The 'quantifiers'

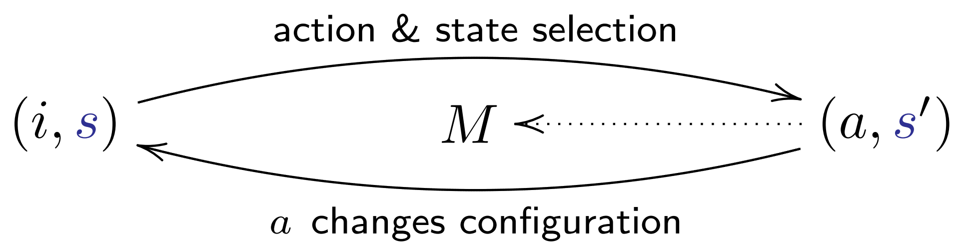
exits ($\exists$)

for all ($\forall$)

ranging over the infinite set $\mathbb{N}$ usually cause undecidability

As any M can be written as $M(n) = UM(p, n)$

actions may be considered also to effect the program p, hence M itself



Memory

Rather than the tape with slow access one can use random access memory

Content addressing gives 'associative memory'

Memory can be also divided into

core	immediately available
disc-cache	quickly available
disc/flash	taking more time

Input/Output

In order to react to the environment some memory cells can be reserved for

values from sensors

values directing actuators

In this way a factory can be controlled by a computer
having access to information from thermometers, barometers
able to turn on/off a heater and exhaust valves

Nature has evolved a chemo-electrical computational model

Synapse Neural Net

These are not programmed using intention but by trial and error

They run in parallel, can be remarkably efficient

Have the same computational power as the Turing Machine model

In human cognition there are zillion possible objects

To program how one reacts to them is undoable

But the following hybrid model [1], [2] has potential

One can see human cognition as a TM with huge I, A sets

in which the transitions are programmed by a NN

Moreover human memory is associative

The configuration can be defined by as the contents of available memory

•	core attention	directly present
•	pre-conscious attention	possible to attend to next cycle
•	unconscious	present in memory and available only later

In the TM the core (cell where the head is) is also available in the next cycle

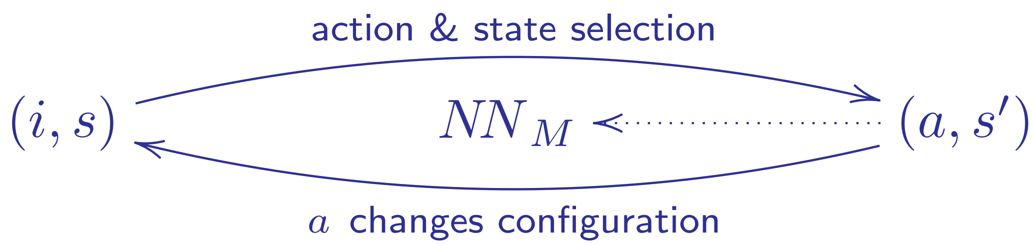
In human cognition better consider core and pre-conscious attention as overlapping

Our conscious attention can be pulled away by a state with desire or aversion

The NN brain has evolved via genes

The TM brain runs with memes, often hampered by states (emotions)

Hybrid Turing model for human cognition [1], [2]



Colleague molecular biology: *"Model also applies to molecular mechanisms:*

discreteness and states (switching on/off genes)
'core-attention' (produced proteins for direct use)
'pre-attention' (prepared proteins for later use)
'un-attention' (potential proteins dormant in genome)"

Discreteness (timescales in ms $10 - 30$, $100 - 300$, $1000 - 3000$ ms)

attentional blink	(10-30)
psychological refractory period	(10-30)
memory search retrieval	(10-30)
wagon-wheel illusion	(100-300)
phenomenology	(1000-3000)

Forms of attention

core-conscious

pre-conscious

un-conscious (‘subliminal conscious’)

Operational definitions given in [0]

States

Mathematical necessity

Overcoming biological noise efficiently [1]

Crucial role for states (including emotions) (i, s)

Need for state-change and state-preservation:

- many neuropeptides (> 50)
- volume transmission (CSF [3], vasopressin, oxytocin [4], β-endorphin [5])

Fundamental instability of perceptible objects (*dukkha*)

Description of meditation

unwholesome	states depending on stabilizing peptides (addiction)
goal	decreasing frequency of unwholesome states
method	sensory restriction
	attentional focus on instantaneous configuration
	concentration (making core- and preconscious attention coincide)
	reflection (taking distance from emotions and other states)
	observing stream of states

[0] Dehaene, Changeux, Naccache, Sackur, Sergent
Conscious, preconscious, and subliminal processing: a testable taxonomy
Trends in Cognitive Sciences, 10(5), 2006, 204-211.

[1] Zylberberg, Dehaene, Roelfsema, Sigman
The human Turing machine: a neural framework for mental programs
Trends in Cognitive Sciences, 2011, 15(7), 293-300.

[2] Barendregt, Raffone
Conscious cognition as a discrete, deterministic, and universal Turing Machine process
Alan Turing - His Work and Impact. Eds. Cooper and van Leeuwen, Elsevier, 2012.

[3] Veening, Barendregt
The regulation of brain states by neuroactive substances distributed via the cerebrospinal fluid. A review *Cerebrospinal Fluid Research*. 7(1), 2010.

[4] Veening, de Jong, Barendregt
Oxytocin messages via the cerebrospinal fluid: behavioral effects; a review
Physiology & Behavior, 101(2), 2010, 193-210.

[5] Veening, Gerrits, Barendregt (In preparation)
β-Endorphin: Effects by volume transmission via the cerebrospinal fluid; a review