What is "Computing"?

Herbert Jaeger

Jacobs University Bremen
Overview

Part 1: Create Confusion:  *Is that Computing?*
Part 2: Call to Order:  *That is Computing!*
Part 3: Seed Doubt:  *Thinking of Matter*
Part 4: Beyond Computing:  *Thinking Matter*
Part 1: Is that Computing?

Note. Sources of pictures and videos are given in separate hidden slides, visible in online version of this talk.
Things that compute (- - really?), selection the first

- IBM PS 2
- HP-35
- Swatch
- Transistor
- Internet (Google server farm)
Things that compute (or what?), selection the second

human brain

physarum polycephalum

English parliament

C. elegans brain
Things that compute (- - eh?), selection the third

- The universe and everything
- Babbage's *difference engine* 2
- Tingueley's *meta-matic* 17 drawing machine
- Pendulum
- Clock
Big questions asked by serious people

• Are "computing", "calculating", "rational reasoning", "information processing", "signal processing" the same?

• Does "intuition", "feeling", "experiencing", "consciousness" give humans some ultimate extra over machines?

• Can physics at the (sub)quantum level be explained in terms of "computing"?
Part 2: *That* is Computing!
The answer is...
Deep historical roots

**Aristotle** (384-322 BC) "All men are mortal. All Greeks are men. Hence, all Greeks are mortal" – Syllogistic logic reasoning as basis of irrefutable reasoning from truth to truth.

**Leibniz** (1646-1716) *Characteristica Universalis* and *Calculus Ratiocinator* – Vision of a universal logical language and mechanical rules of argumentation

**Boole, Frege, Hilbert, Russell, Gödel** (~1850'ies to 1930'ies) *mathematical logic systems* – dream of mechanically deciding all mathematical questions
THE question answered by Turing with Turing machines: Hilbert's *Entscheidungsproblem*

Is there an effective procedure for deciding every mathematical conjecture?

\[
\text{Axioms} \Rightarrow \text{Claimed Theorem}
\]

\[
\{a_1, \ldots, a_n\} \Rightarrow c
\]

\[
a_1\#\ldots\#a_n\#c \leftrightarrow \{\text{yes, no}\}
\]

\[
<\text{codeNr of } a_1\#\ldots\#a_n\#c > \leftrightarrow \{0, 1\}
\]

Is there an effective procedure for computing every finitely definable function on the natural numbers?
Is there an effective procedure for computing every finitely definable function on the natural numbers?

• Turings approach: investigate and formalize what a human "computer" can and cannot do with the aid of his brain and paper and pencil

• Quotes on next slides are from Turing's groundbreaking 1936 paper

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM
"Computing is normally done by writing certain symbols on paper. [...] I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent $j$. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols."

- An "effective procedure" uses finite set of identifiable symbols (an alphabet).
- There must be an unlimited supply of "paper" (computer memory) for storing arbitrarily long symbol sequences.
"The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. [...] We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused."

- The apparatus ("mind") which processes symbolic information is a finite state-switching system.
The Turing Machine (TM)

Is there a finite state-switching machine commanding on an unbounded symbol-sequence memory, which can compute every finitely definable function on the natural numbers?

**Transition rule table:**

- \((S, \rightarrow) \rightarrow (r, \rightarrow, \rightarrow)\)
- \((r, 0) \rightarrow (t, 1, -)\)
- ... 
- \((t, 1) \rightarrow (H, 1, \rightarrow)\)

**Tape alphabet:** \{\(\rightarrow\), 0, 1, -\}

**Finite state set, including start and halt state:** \{\(S, r, t, H\)\}
**Demo**: compute function $f(n) = 2n$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>-</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>$\gg$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$r$</td>
<td>$\gg$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$r$</td>
<td>$\gg$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$r$</td>
<td>$\gg$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$H$</td>
<td>$\gg$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Start**: input $n = 5$ (binary)

- rule: $(S, \gg) \rightarrow (r, \gg, \rightarrow)$
- rule: $(r, 1) \rightarrow (r, 1, \rightarrow)$
- rule: $(r, 0) \rightarrow (r, 0, \rightarrow)$
- rule: $(r, 1) \rightarrow (r, 1, \rightarrow)$
- rule: $(r, -) \rightarrow (H, 0, -), \text{halt!}$
Turing Machines, comments

• TMs can compute all functions that your PC can compute

• Church-Turing (-Rosser) hypothesis: *Every function that can be computed by whatever "effective" mechanism can also be TM-computed*

• Two main arguments in favor of this hypothesis:
  1. so far, no counterexample
  2. many other formalizations of "effective computing mechanism" proven equivalent to TMs

• Turing found a finitely definable function $f$ that he proved is not TM-computable: the *halting problem*

• This gives a negative answer to Hilbert's Entscheidungsproblem: *there is no effective procedure to decide all mathematical conjectures*
Cellular automata (CA), another model of universal computation

1-dimensional CA, by example

\[
\begin{array}{cccccccc}
\cdots & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \cdots & t &= 0 \\
\cdots & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \cdots & t &= 1 \\
\cdots & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & \cdots & t &= 2 \\
\cdots & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & \cdots & t &= 3 \\
\cdots & & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{array}
\]
Cellular automata, core properties

• Like TMs:
  - finite symbol set (memory symbols and state symbols coincide)
  - discrete time "computing steps"
  - update rules in finite table
  - unbounded memory

• CA first introduced by von Neumann (and others, 1950'ies) as model for biology-inspired self-reproducing systems
  - maybe first appearance of theme "life =? computing"

• CA can compute exactly the same functions $f: \mathbb{N} \mapsto \mathbb{N}$ as TMs

• Use in Computer Science: basic model of parallel computing
CAs for modeling nature

• for 1-dim, 2-dim, ... systems
• model for spatio-temporal dynamics
• PDE models can be approximated by CAs via discretization
• Popular to model pattern formation
• Popular to analyze self-organization classes in spatiotemporal systems
Resublimation

Nonlinear wave dynamics

Population dynamics

Reaction-diffusion systems
It is the only known, and unchallenged, and universal clarification of effective computing. THAT is Computing!

Btw., quantum computers cannot compute more functions than TMs. Proposals for hypercomputing "mechanisms" need infinite precision, or infinite speedup, etc. - and are defined relative to Turing computability.

Modern philosophy of computation revolves around Turing computability. Pancomputationalism revolves around Turing computability.
Part 3: Thinking of Matter
Riddle

• What general-purpose computer came before Turing Machines?

• What computer is hosting the theory of computing?

• What computer can do more and do less than TMs, at the same time?

• What singular computing system receives the largest public research funding?
Two worlds of computing

WHEREIN is computing realized? – the **material substrates**

- transistors, copper wires, solder, batteries
- nerve cells, glia cells, vessels, neurotransmitters, hormones

HOW is it computed? – the **"algorithmic" procedures**

- symbol pattern matching, table lookup, symbol replacement
- spike pulses, neurotransmitter chemistry, nonlinear neural field dynamics

WHAT is computed? – the **tasks**

- functions, simulations, visualizations, digital signals, internet host connections, ... *everything?*
- sonnets, breathing, walking, jokes, tears, tax declarations, nonsense, dreams..., *everything?*

*Let me not to the marriage of true minds admit impediments*
Is it the same?

Q: Can human, brain-based intelligence be explained by / reduced to Turing computation?

A: Yes and no.
The YES IT IS THE SAME faction

"A physical symbol system has the necessary and sufficient means for general intelligent action. [...] By "general intelligent action" we wish to indicate the same scope of intelligence as we see in human action: that in any real situation behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur, within some limits of speed and complexity." (From the Turing Award Lecture of Allen Newell and Herbert A. Simon, 1975)

• Explicitly construed as equivalent to universal TMs
• Claimed evidence for sufficient: progress in AI
• Claimed evidence for necessary: progress in cognitive psychology
Some NO IT'S NOT THE SAME factions

Epistemology

- Scattering of the *same* concept ("explain", "reduce", "functional equivalence", "ontological identity")

Philosophy of mind

- Consciousness, experience of qualia, mental attitudes not captured by symbolic computation

Connectionism

- Information representation and processing in parallel, distributed systems (neural networks) is not symbolic

Behavior-based robotics, "New AI"

- Biological intelligence is situated and embodied

Evolutionary theories of intelligence

- Rational reasoning and symbolic language emerged from pre-rational intelligence
Representing discrete symbols by analog quantities

- Core question for Physical Symbol System hypothesis
- Important question for unconventional computing / unconventional substrates
Symbols in dynamical systems, examples

• Bistability
  - digital circuits are analog at heart
  - bistable neurons suspected in prefrontal cortex, associated with working memory

• Multistable systems
  - THE classical model of neural memory: Hopfield networks
  - combined with stochastic sampling dynamics: Boltzmann machines – historical (2006) quickstarter of deep learning
Symbols in dynamical systems, more examples

- Periodic attractors
  - cyclic rehearsal: Baddeley model of working memory
  - rich theory: coupled oscillator systems
  - options to couple together symbolic items

- Chaotic attractors
  - Freeman model of representing odors in olfactory bulb
  - options for representing hierarchical symbolic structures
Symbols in dynamical systems, even more examples

- Saddle note dynamics
  - heteroclinic channel model of cognitive dynamics (Rabinovich 2008)
  - option to explain transient stability

- Spike time patterns
  - e.g. *polychronicity* model of Izhikevich (2006)
  - options to exploit fine-grained neural information
Symbols in dynamical systems, ever more examples

• Conceptors
  – switchable dynamical modes of recurrent neural networks
  – temporal long term memory model for dynamical patterns
  – options for neural coding of conceptual structures

• Spatial coding
  – the grand alternative to attractor-like phenomena
  – "grandmother cells"
  – what you see in neural imaging
Mind, math and matter: summary comments

• Symbols in neural dynamics?
  - too many candidates for single definite answer
  - hence, unclear whether / in what sense brains "compute"

• Impact on engineering!
  - novel materials share core properties with brain wetware:
    ✴ analog
    ✴ unclocked
    ✴ parallel
    ✴ stochastic
    ✴ (very) low precision
    ✴ drift, aging, irreproducibility

hardly compatible with TM model
My opinion

Essence of TM computing
Combine few "atomic", static symbols into arbitrarily large, arbitrarily complex data structures

Essence of brain computing
Combine limited small number (7 ± 2) of very complex, dynamical concept instantiations into "organic" states of mind
Part 4: Thinking Matter
In the Whirlpool
Where to go from here

Where we stand:
- TM model only partially suitable for capturing the brain's information processing (IP)
- unconventional computing and substrates are similar to brains in important ways

What we wish to have:
- insightful guidance for practical unconventional IP engineering

How to move on:
- take stock of existing alternative scientific IP paradigms
- identify key design coordinates for unconventional IP machines
Alternative paradigms with promises for brainlike IP

Signal processing and control

**Where**: engineering, neuroscience  ● **Objects**: time series data, sensor signals, motor commands  ● **Math**: linear systems, information theory  ● **Why brainlike**: ... cybernetics!

Nonlinear optimization, stochastic search

Operations research, theoretical CS, theoretical physics  ● Complex optimization problems  ● Variational calculus, sampling techniques, statistical thermodynamics, unsupervised and reinforcement learning  ● Free energy model of autonomous agent (Friston); stochastic parallel dynamics

Self-organizing dynamical systems, pattern formation

Theoretical physics, theoretical biology  ● Spatiotemporal nonlinear systems  ● PDEs, field dynamics, coupled oscillators, phase transitions  ● Neural field theories; attractor models of concept representations
Brainlike IP (ahem... how's that work?) – design coordinates

• We perceive some "negative" global characteristics
  – like, stochastic, analog, unclocked, low-precision, ...

• (Computational) neuroscience doesn't give many exploitable hints
  – no system blueprint
  – some local mechanisms, like STDP, cochlear model for acoustic processing, ...

What would be sections in a brainlike IP engineering textbook?
# Table of Contents

1 Mathematical Preliminaries

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Signals and Systems</td>
<td>1</td>
</tr>
<tr>
<td>Stochastic Processes</td>
<td>53</td>
</tr>
<tr>
<td>Information Theory</td>
<td>87</td>
</tr>
<tr>
<td>Nonlinear Dynamics</td>
<td>112</td>
</tr>
<tr>
<td>Graphs and Networks</td>
<td>145</td>
</tr>
<tr>
<td>Brainlets</td>
<td>155</td>
</tr>
</tbody>
</table>
2 Signal Propagation

Wired Transmission 180
Soliton Dynamics 193
Diffusion and Reaction-Diffusion Processes 232
2- and 3-Dimensional Field Dynamics 262
Electrons, Photons, Mechanons Chemons, and their Interactions 314
3 Timing

Timescales 344
Synchronization and Delays 345
Slowing Down and Speeding up 372
Time vs. Space 392
Timescale Cocktails 456
4 Persistent Memory

Very-long Timescale Phenomena 477
Discrete vs. Random vs. Analog Measurables 489
Physical Endurance and Size 501
Aging 528
Distributed Memory 561
5 Transient Memory

State Dimension and Representation Capacity 590

Attractors vs. Transients 612

1/p Processes 640

Echo States 700
6 Energy

Thermodynamical Limits 750

Dissipation and the Creation of Information 769

"Zero-Energy" Processes and Ambient IP 812

Adiabatic computing 833
# 7 Information Coding

Structure in Time and Space 850  
Is Encoding Necessary? 858  
Decoding vs. Interpretation 881  
Nonstandard Measures of "Information" 920  
Content Addressing 943  
Addressing by Generalized Location 988  
Coupling and Binding 1003  
Multiscale Representation Hierarchies 1043
8 Robustness

Stability Measures in Input-Driven Systems 1061

Robust Control 1093

Growth and Structural Reorganization 1128

Classical Homeostasis: Operational Stabilization 1160

Generalized Homeostasis: Functional Stabilization 1193

References 1231-1299
Functional stabilization – done by a real brain

Ivo Kohler and Theodor Erismann at the University of Innsbruck, Austria, ~1950
Functional stabilization with conceptors: demo

Task specification

**Given:** transform signal $F(s(n))$ obtained from passing reference signal $s(n)$ through some filter $F$.

**Wanted:** equalizing RNN $H$ which transforms $F(s(n))$ back to $s(n)$: $H(F(s(n))) \approx s(n)$

We want much more!

- The same equalizing RNN $H$ should work for *any* transform $F$!
- This generic equalizer $H$ should be trained single-shot on the reference $s(n)$ as *only* training data!

Idea

- The equalizing RNN $H$ has *homeostatic dynamics*: it tries to self-regulate to attain a reference RNN dynamics envelope (shape of state space described by reference conceptor $c^{ref}$)
- A self-regulation mechanism inside $H$ tries to pull the state space geometry toward $c^{ref}$. 
Results

• Reference $s(n)$ is sum of two incommensurable sines

• Neural homeodynamics $H$: 5-stage cascade, each stage with 50 + 200 neurons

$F$: shift and upscaling

$F$: nonlinear autoregression

$F$: additive noise, SNR = 1
Summary
Sources

Things that compute 1
HP calculator: http://www.hpmuseum.org/hp35.htm
Transistor: http://www.robotpark.com/Transistor-En
Google Server Farm: https://abbyherbert.wordpress.com/tag/google-server-farms/

Things that compute 2
Physarum polycephalum: https://www.youtube.com/watch?v=8IRKmCUa2N0
UK parliament: http://www.parliament.uk/about/mps-and-lords/members/

Things that compute 3
Tinguely drawing machine: https://www.pinterest.com/pin/2828303880163976/
Clock: http://bestanimations.com/HomeOffice/Clocks/Grandfather/Grandfather.html
Pendulum: https://www.questacon.edu.au/qshop/Foucault-s-Pendulum/
Universe: https://www.sciencedaily.com/releases/2016/06/160622144930.htm

The answer is...
Boulder: https://www.nhstateparks.org/visit/state-parks/madison-boulder-natural-area.aspx

Deep historical roots
Aristotle: https://www.britannica.com/biography/Aristotle
Leibniz: https://www.mathematik.ch/mathematiker/leibniz.php
Boole: https://commons.wikimedia.org/wiki/File:Portrait_of_George_Boole.png
Frege: https://en.wikipedia.org/wiki/Gottlob_Frege
Russell: https://en.wikipedia.org/wiki/Bertrand_Russell
Gödel: https://www.britannica.com/biography/Kurt-Godel
CAs for modeling nature

CS simulation videoclips
http://www.fourmilab.ch/cellab/ (Rudy Rucker and John Walker)

Polaris starry night
https://stargazerslounge.com/topic/142322-polar-star-trails/

Two worlds of computing
cpu: http://www.kantankeizai-kumaburogu.com/entry/2017/06/09/151452
spike trains: https://www.researchgate.net/figure/10822994_fiq12_FIG-12-Spike-train-patterns-of-the-neuron-subjected-to-different-levels-of-noise-input
"let me not..." Shakespeare Sonnet 116
Excel snapshot: https://support.office.com/ms-my/article/Gambaran-keseluruhan-jadual-Excel-7ab0bb7d-3a9e-4b56-a3c9-6c94334e492c

Representing discrete symbols by analog quantities
EEG plot: https://www.physionet.org/pn6/chbmit/

Symbols in dynamical systems, examples
Hopfield network: http://fourier.eng.hmc.edu/e161/lectures/nn/node5.html

Symbols in dynamical systems, even more examples
heteroclinic path: from Rabinovich et al (2008), Transient Cognitive Dynamics, Metastability, and Decision Making, PLOS Comp. Biol. 4(5), e1000072
cronychronous groups: from Izhikevich, E. M. (2006), Polychronization: Computation with Spikes, Neural Computation 18, 245-282

Symbols in dynamical systems, even more examples

My opinion
mathematician: http://www.cam.ac.uk/research/features/does-economics-need-less-maths-or-more
computer with operator: http://listelist.com/alan-turing-kimdir/

Where to go from here...

Brainlike IP – ahem... how's that work?
textbook: http://www.wikihow.com/Read-a-Textbook
The project

Table of contents slides
student 1: https://azsiaekkovei.hu/index/2
student 2: https://www.testingmom.com/blog/olsat-level-b-2/
student 3: http://www.thewestwoodvillage.com/services/mentor-language-institute
student 4: https://www.cowinmusic.com/blog/can-noise-cancelling-headphones-improve-student-exam-results/
student 7 (flying books): https://www.pinterest.com/mattesonlibrary/flying-books/
student 8 (kids):

Functional stabilization – done by a real brain
fencing clip: https://www.youtube.com/watch?v=C-Opnrb6I

Last slide
Bleriot departing from France: https://www.gracesguide.co.uk/File:Im1909FL-Bleriot04.jpg