

Title: When does a physical systems compute: Is physics more or less than computation?

Date: Apr 13, 2018 11:30 AM

URL: <http://pirsa.org/18040108>

Abstract: Landauer's famous dictum that 'information is physical' has been enthusiastically taken on by a range of communities, with researchers in areas from quantum and unconventional computing to biology, psychology, and economics adopting the language of information processing. However, this rush to make all science about computing runs the risk of collapsing into triviality: if every physical process is computing, then to say that something performs computation gives no meaningful information about it, leaving computational language devoid of content. In this talk I will give an introduction to Abstraction/Representation Theory, a framework for representing both computing and physical science that allows us to draw a meaningful distinction between them. The use of AR theory - with its commuting-diagrammatic framework and associated algebra of representation - allows us to take significant steps towards giving a formal language and framework for the processes of science. I will show how AR theory represents this process (including the potential for automation), and the insights it gives into the usage and limits of computation as a formal process language for, and description of, physical sciences.

***When does a physical system compute:
Is physics more or less than computation?***

Dominic Horsman

In collaboration with Susan Stepney (York) and Viv Kendon (Durham)

*'Algorithmic information, induction,
and observers in physics'*

Perimeter Institute

13 April 2018





Outline: physics and computing

The language of information theory is frequently used (to a greater or lesser degree of precision) in physical theories.

Can it be made precise? How do physics and computation relate?

The starting point: computing is physical.

Abstraction/representation theory: a framework for reasoning about representation (interface between physical and abstract systems).

Representation -> modelling -> scientific theories -> computing

Observers in AR theory: 'representational entities'.

Intrinsic logics: a correct use of computational language for physical systems.

References

When does a physical system compute?

C. Horsman, S. Stepney, R. Wagner, V. Kendon
Proc. R. Soc. A 2014 470 20140182 July 2014

References

When does a physical system compute?

C. Horsman, S. Stepney, R. Wagner, V. Kendon
Proc. R. Soc. A 2014 470 20140182 July 2014

Abstraction/Representation Theory for heterotic physical computing D. Horsman
Phil. Trans. R. Soc. A 2015 373 20140224; DOI: 10.1098/rsta.2014.0224 (2015)

The representation of computation in physical systems D. Horsman
in EPSA15 Selected Papers, proceedings of the 5th conference of the European
Philosophy of Science Association, Springer (2017)

The Natural Science of Computing D. Horsman, V. M. Kendon, S. Stepney
Communications of the ACM 60(8) 31-34 (2017)

Abstraction and representation in living organisms: when does a biological system compute? D. Horsman, V. Kendon, S. Stepney, P Young
In G. Dodig-Crnkovic and R. Giovagnoli (Eds), Representation and Reality: Humans, Animals, and Machines, Springer 2018.

The Bandwagon

CLAUDE E. SHANNON

INFORMATION theory has, in the last few years, become something of a scientific bandwagon. Starting as a technical tool for the communication engineer, it has received an extraordinary amount of publicity in the popular as well as the scientific press. In part, this has been due to connections with such fashionable fields as computing machines, cybernetics, and automation; and in part, to the novelty of its subject matter. As a consequence, it has perhaps been ballooned to an importance beyond its actual accomplishments. Our fellow scientists in many different fields, attracted by the fanfare and by the new avenues opened to scientific analysis, are using these ideas in their own problems. Applications are being made to biology, psychology, linguistics, fundamental physics, economics, the theory of organization, and many others. In short, information theory is currently partaking of a somewhat heady draught of general popularity.

Although this wave of popularity is certainly pleasant and exciting for those of us working in the field, it carries at the same time an element of danger. While we feel that information theory is indeed a valuable tool in providing fundamental insights into the nature of communication problems and will continue to grow in importance, it is certainly no panacea for the communication engineer or, *a fortiori*, for anyone else. Seldom do more than a few of nature's secrets give way at one time. It will be all too easy for our somewhat artificial prosperity to collapse overnight when it is realized that the use of a few exciting words like *information*, *entropy*, *redundancy*, do not solve all our problems.

What can be done to inject a note of moderation in this situation? In the first place, workers in other fields should realize that the basic results of the

subject are aimed in a very specific direction, a direction that is not necessarily relevant to such fields as psychology, economics, and other social sciences. Indeed, the hard core of information theory is, essentially, a branch of mathematics, a strictly deductive system. A thorough understanding of the mathematical foundation and its communication application is surely a prerequisite to other applications. I personally believe that many of the concepts of information theory will prove useful in these other fields—and, indeed, some results are already quite promising—but the establishing of such applications is not a trivial matter of translating words to a new domain, but rather the slow tedious process of hypothesis and experimental verification. If, for example, the human being acts in some situations like an ideal decoder, this is an experimental and not a mathematical fact, and as such must be tested under a wide variety of experimental situations.

Secondly, we must keep our own house in first class order. The subject of information theory has certainly been sold, if not oversold. We should now turn our attention to the business of research and development at the highest scientific plane we can maintain. Research rather than exposition is the keynote, and our critical thresholds should be raised. Authors should submit only their best efforts, and these only after careful criticism by themselves and their colleagues. A few first rate research papers are preferable to a large number that are poorly conceived or half-finished. The latter are no credit to their writers and a waste of time to their readers. Only by maintaining a thoroughly scientific attitude can we achieve real progress in communication theory and consolidate our present position.

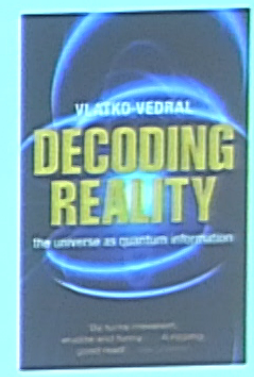
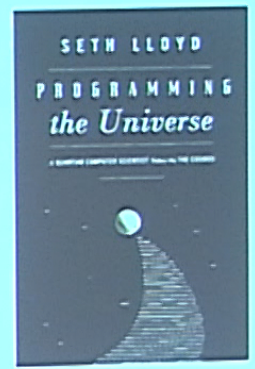
INFORMATION theory has, in the last few years, become something of a scientific bandwagon.

beyond its actual accomplishments. Our fellow scientists in many different fields, attracted by the fanfare and by the new avenues opened to scientific analysis, are using these ideas in their own problems. Applications are being made to biology, psychology, linguistics, fundamental physics, economics, the theory of organization, and many others. In short, information theory is currently partaking of a somewhat heady draught of general popularity.



A 2012 parody of the Shannon bandwagon from American electrochemical engineer Libb Thims' article "Thermodynamics ≠ Information Theory: Science's Greatest Sokal Affair." [3]

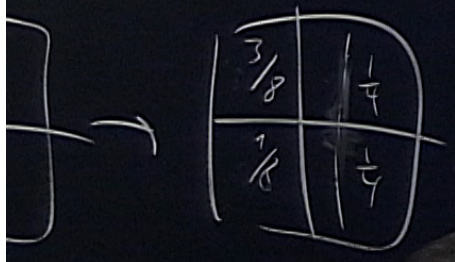
"The universe is a (quantum) computer:"



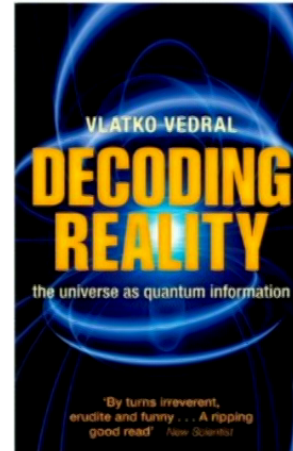
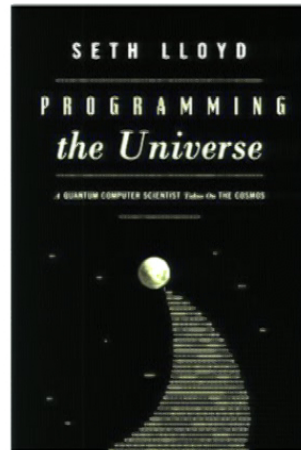
DAVID J. CHALMERS

DOES A ROCK IMPLEMENT EVERY FINITE-STATE AUTOMATON?

ABSTRACT: Hilary Putnam has argued that computational functionalism cannot serve as a foundation for the study of the mind, as every arbitrary open physical system implements every finite-state automaton. I argue that Putnam's argument fails, but that it points out the



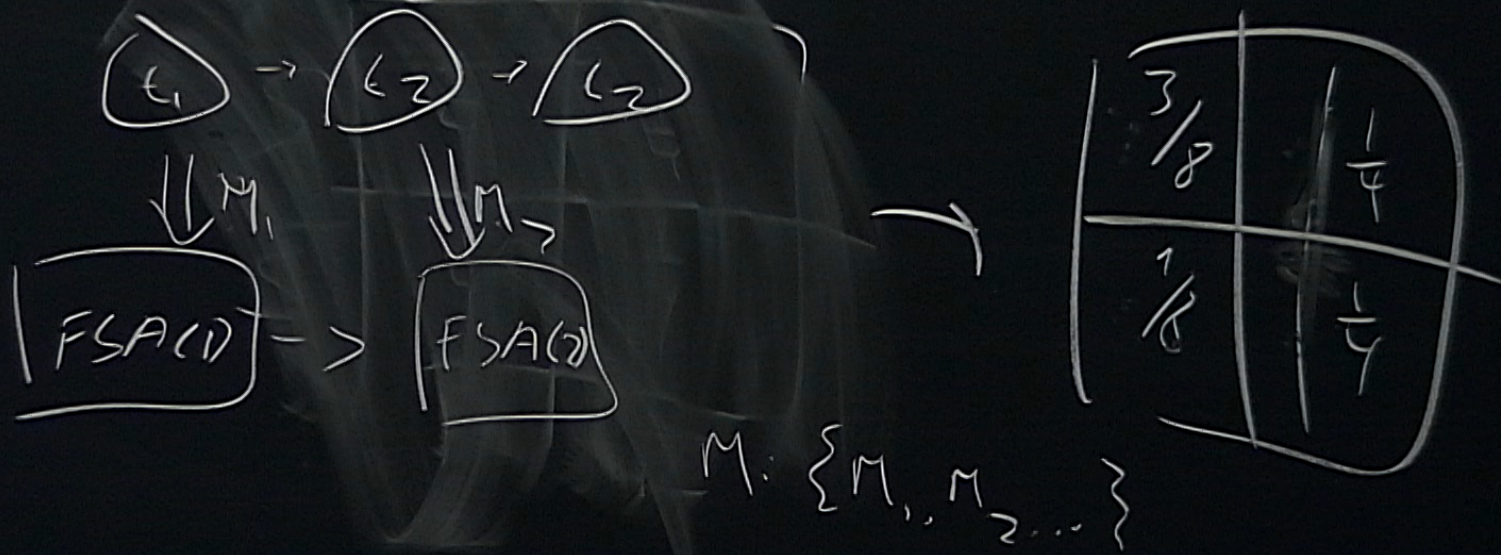
“The universe is a (quantum) computer:”



DAVID J. CHALMERS

DOES A ROCK IMPLEMENT EVERY FINITE-STATE
AUTOMATON?

ABSTRACT. Hilary Putnam has argued that computational functionalism cannot serve as a foundation for the study of the mind, as every ordinary open physical system implements every finite-state automaton. I argue that Putnam's argument fails, but that it points out the need for a better understanding of the bridge between the domain of computation and the



How can we put rigorous meaning behind the use of
information-theoretic language in physics?
(Never mind everywhere else...)

Two extremes to avoid:

Everything computes all the time

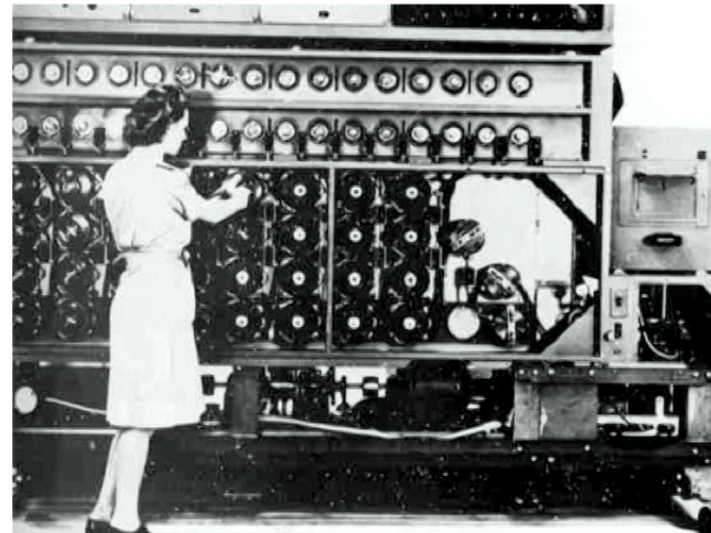
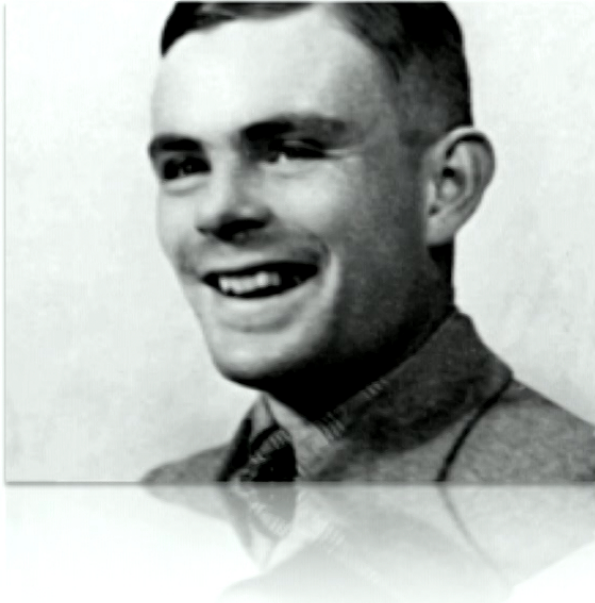
Computing requires consciousness/other magic

If there is physical content to the assertion “system X computes” then we need a better way of telling when it does.

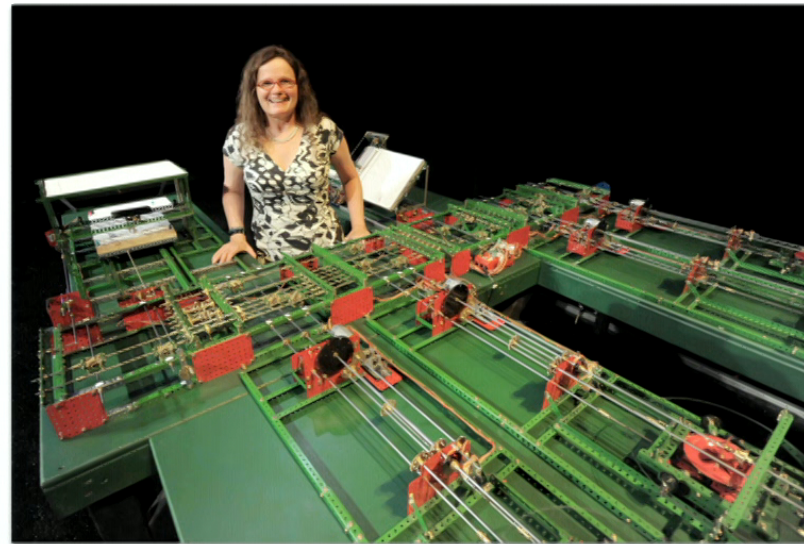
So: when **does** a physical system compute?
What does physics have to do with computation in the first place
(isn't that just a branch of mathematics?)

Proposal: computer science is the natural science of the
computing abilities of physical systems

Computers and CS theory: Turing and the Bombe



Computers and CS theory: Shannon and the Differential Analyser



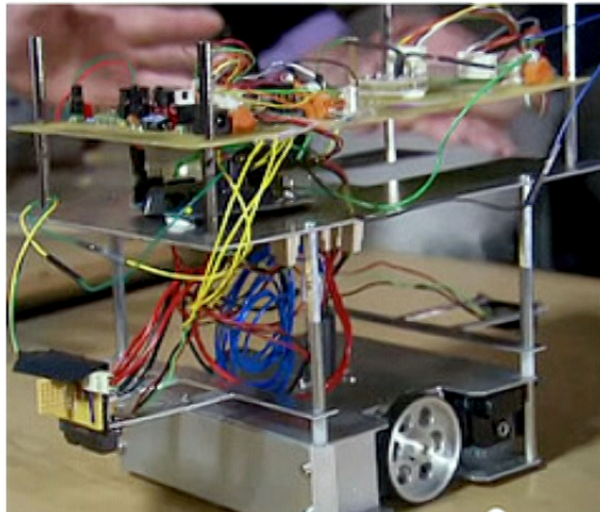
Dr Bonita Lawrence
(Marshall University)

What do we do with this?



IBM Q 50 qubit
device

Or this?



“PhiBot”
Slime-mould
controlled robot
Zauner Group
Southampton

www.sense.ecs.soton.ac.uk/

Or this?



Openworm Project
Lego robot
controlled by
simulated *C. Elegans*
neurons

www.openworm.org

Or this?

Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

Adaptive Flight Control With Living Neuronal Networks on Microelectrode Arrays

Thomas B. DeMarse and Karl P. Dockendorf

Department of Biomedical Engineering, University of Florida, Gainesville, Florida.

Abstract- The brain is perhaps one of the most robust and fault tolerant computational devices in existence and yet little is known about its mechanisms. Microelectrode arrays have recently been developed in which the computational properties of networks of living neurons can be studied in detail. In this paper we report work investigating the ability of living neurons to act as a set of neuronal weights which were used to control the flight of a simulated aircraft. These weights were manipulated via high frequency stimulation inputs to produce a system in which a living neuronal network would "learn" to control an aircraft for straight and level flight.

I. Introduction

Research into the computational properties of living neuronal networks has seen a rapid explosion in interest of the last two decades. This interest has been fostered by the advent of technology able to simultaneously measure neural activity from hundreds of neurons both in vivo [1-3] and in vitro [4-8]. However, many of the computational properties exhibited by these networks remain unclear.

Our approach is to use a system where we can measure, stimulate, and therefore manipulate activity across a grid of 60

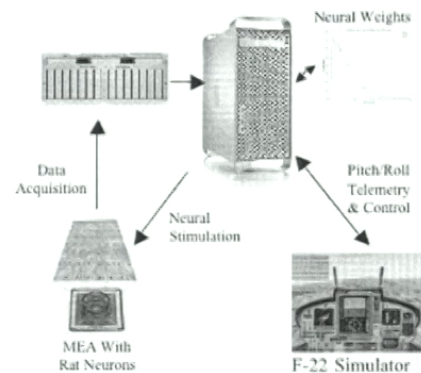


Fig. 1. Schematic of Neural Flight Control System using living rat cortical neurons for pitch and roll control.

Or this?

Rats on a plane!

Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

Adaptive Flight Control With Living Neuronal Networks on Microelectrode Arrays

Thomas B. DeMarse and Karl P. Dockendorf

Department of Biomedical Engineering, University of Florida, Gainesville, Florida.

Abstract- The brain is perhaps one of the most robust and fault tolerant computational devices in existence and yet little is known about its mechanisms. Microelectrode arrays have recently been developed in which the computational properties of networks of living neurons can be studied in detail. In this paper we report work investigating the ability of living neurons to act as a set of neuronal weights which were used to control the flight of a simulated aircraft. These weights were manipulated via high frequency stimulation inputs to produce a system in which a living neuronal network would "learn" to control an aircraft for straight and level flight.

I. Introduction

Research into the computational properties of living neuronal networks has seen a rapid explosion in interest of the last two decades. This interest has been fostered by the advent of technology able to simultaneously measure neural activity from hundreds of neurons both in vivo [1-3] and in vitro [4-8]. However, many of the computational properties exhibited by these networks remain unclear.

Our approach is to use a system where we can measure, stimulate, and therefore manipulate activity across a grid of 60

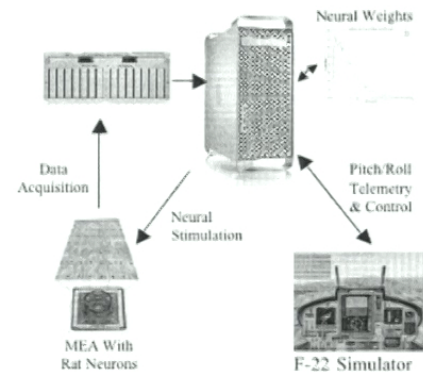
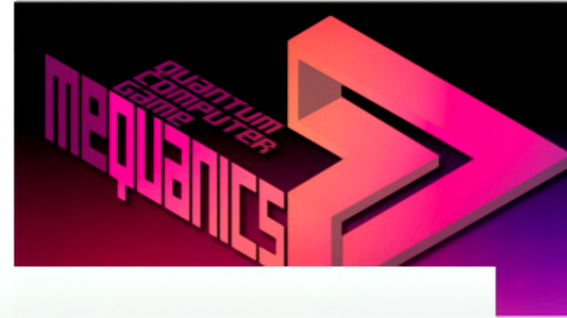
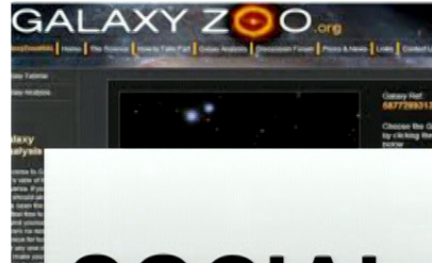


Fig. 1. Schematic of Neural Flight Control System using living rat cortical neurons for pitch and roll control.

What else?



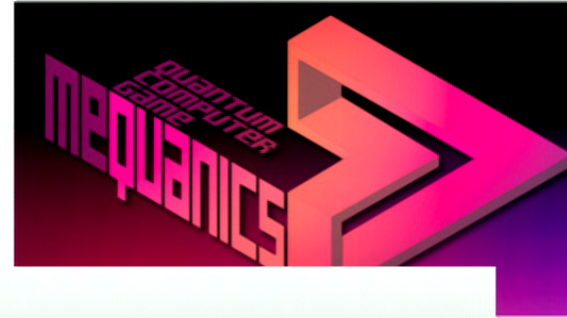
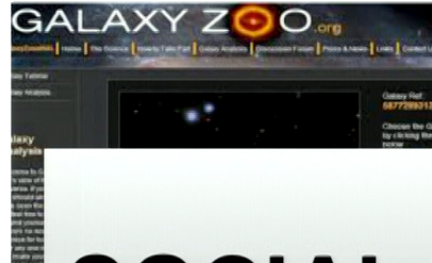
SOCIAL MACHINES



WIKIPEDIA
The Free Encyclopedia



What else?



SOCIAL MACHINES



WIKIPEDIA
The Free Encyclopedia



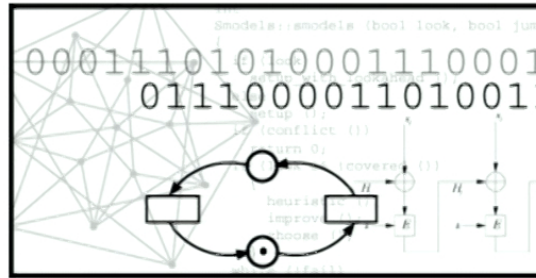
sociam.org

Abstract



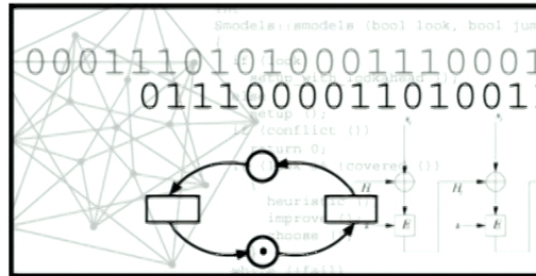
Physical

Abstract



Physical

Abstract

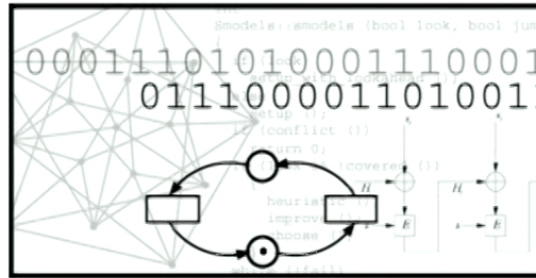


.....

Physical



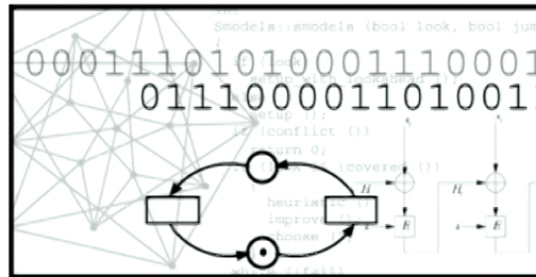
Abstract



Physical



Abstract

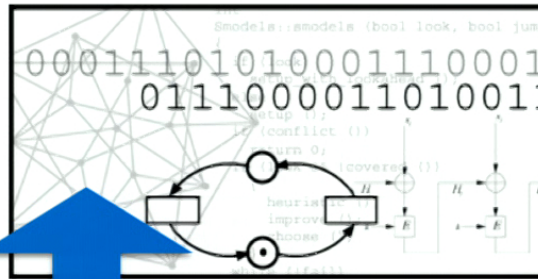


Physical



“Just engineering”?

Abstract



.....

AR THEORY

Physical



Representation

The central issue: how do we get from the physical to the abstract domain?

Answer: REPRESENTATION.

The core of Abstraction/Representation Theory is the **representation relation** between physical and abstract objects.

Representation

Abstract



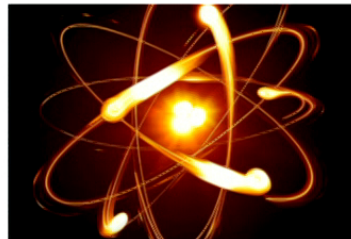
Physical

Representation

Abstract



Physical



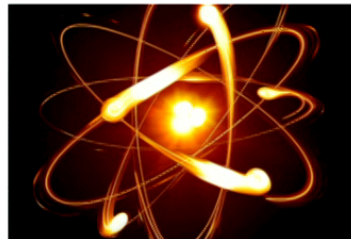
Representation

Abstract

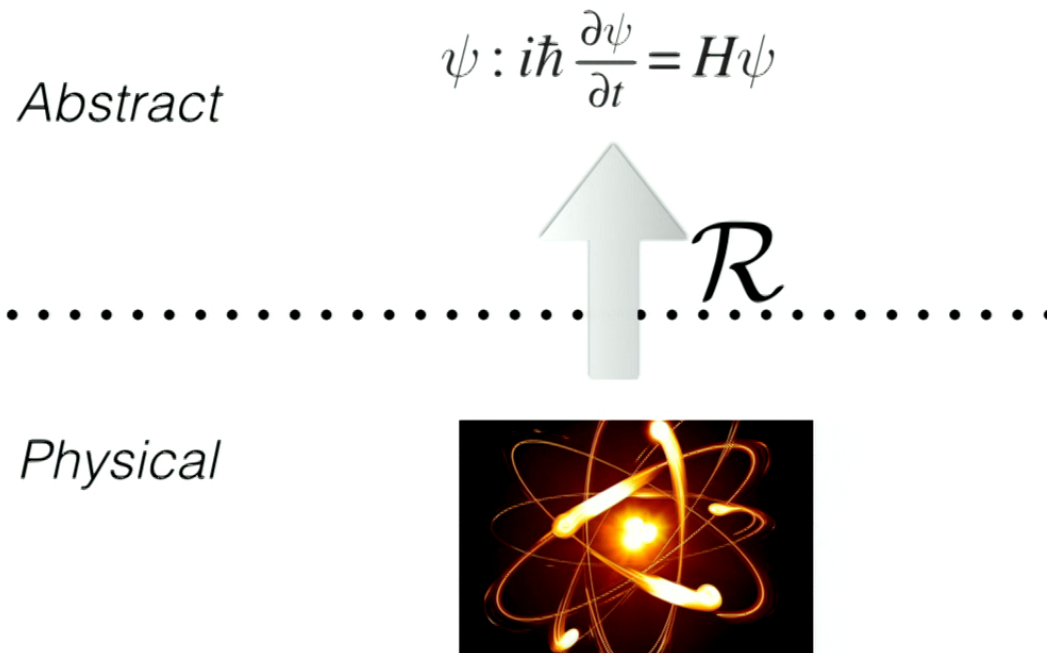
$$\psi : i\hbar \frac{\partial \psi}{\partial t} = H\psi$$



Physical



Representation



Representation

What is representation and what is it not?

The representation relation R maps physical to abstract objects.

This is **not a mathematical function**.

This is **not a logical relation**.

What is it?

Good question! But we know it exists, so let's interrogate it.

Representational issues appear in **a wide range of areas**.

Representation

Brief context:

Rorty (1979) *Philosophy and the Mirror of Nature*

- Representation is not mirroring of physical in abstract

van Fraassen (2008) *Scientific Representation*

- The representation is not the thing, but encodes scientific theories

See also:

Carnap (1928) *Aufbau (The Logical Structure of the World)*

- The universal objects of science are sense-data (repudiated)

See also also:

Wittgenstein (early), Putnam, Fodor, Frigg, Quine, Hartmann...

Note: starting from a **physical** perspective of computing, not a **semantic/mental** one.

Representation

Aren't there already representations of computers in theoretical CS?

Machine code, words, concrete semantics, etc?

Isn't a "computer" a model of a formal system?

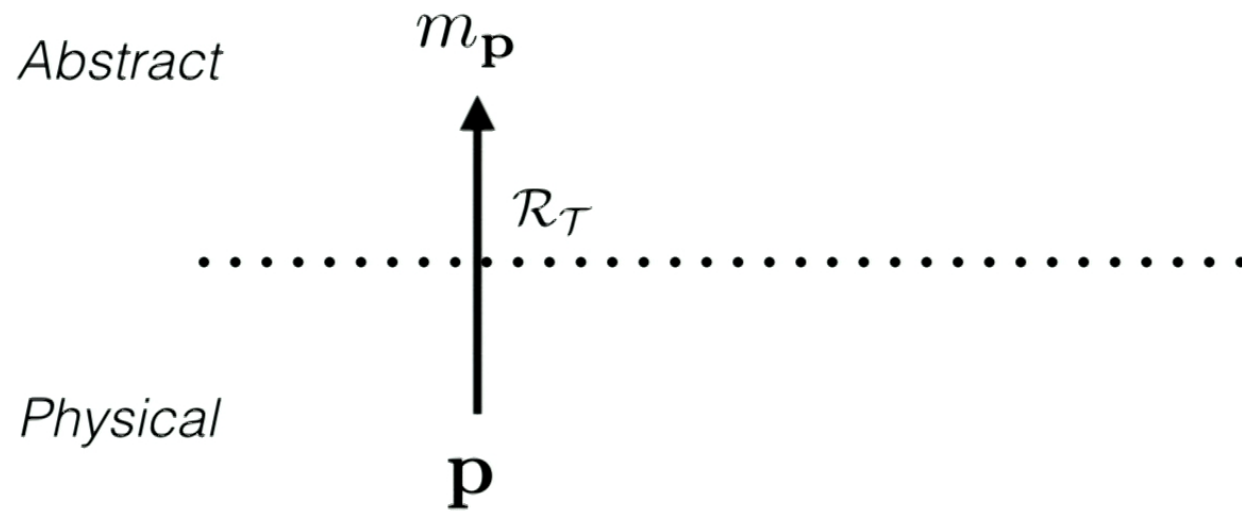
No:

A model or a code or a semantics is still an abstract object, not the physical computer itself.

When we talk about **p** we mean the physical system itself, not a representation of it.

AR Theory

Fundamental representation:



AR Theory

In algebraic terms:

Domain of physical objects \mathbf{P}

Domain of abstract objects M

Directed relation $\mathcal{R} : \mathbf{P} \rightarrow M$

For each $\mathbf{p} \in \mathbf{P}$

if \mathcal{R} takes $\mathbf{p} \rightarrow m \in M$

then m is written $m_{\mathbf{p}}$

and $\langle \mathbf{p}, \mathcal{R}, m_{\mathbf{p}} \rangle$

forms a **representational triple**.

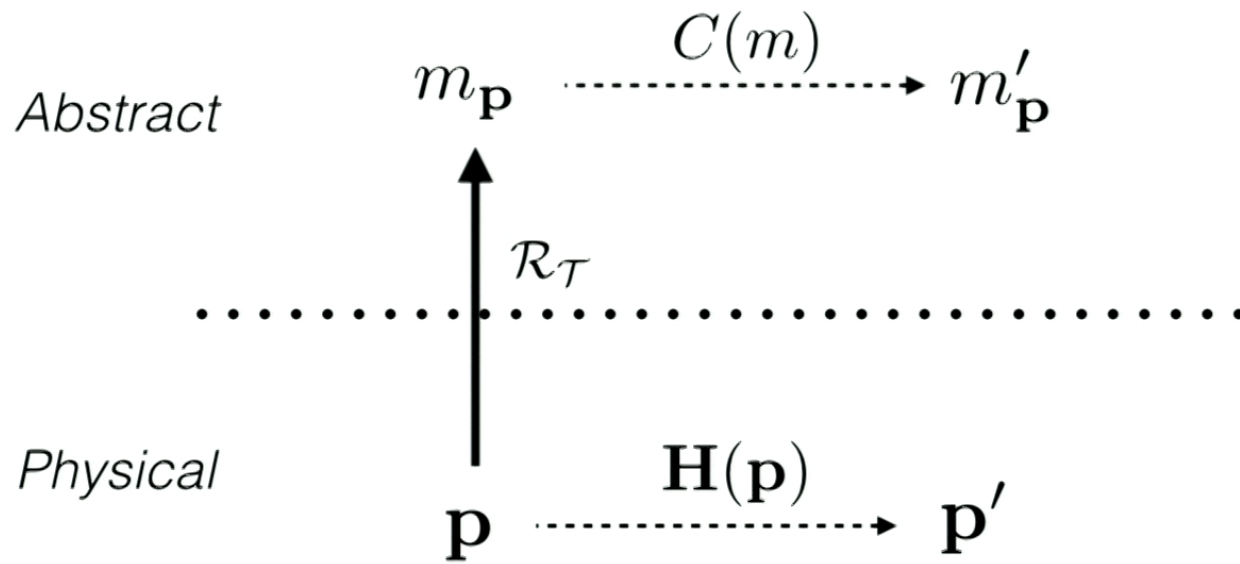
AR Theory

This is the **modelling relation**.

What makes something a good model? This is (partly) the domain of experimental science:

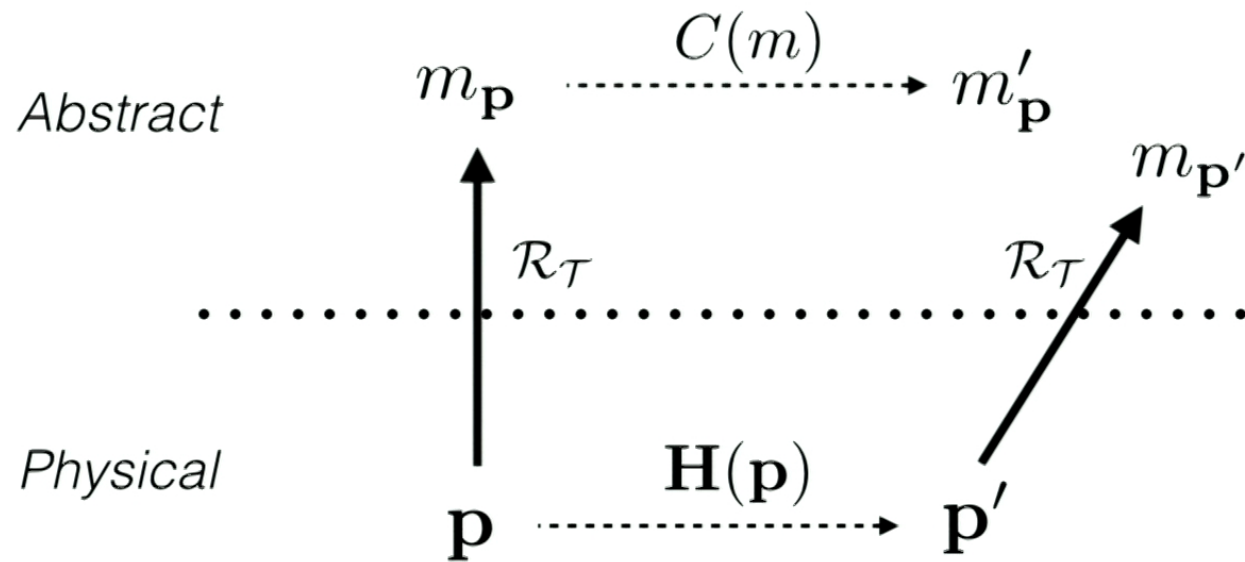
AR Theory

Abstract evolution:



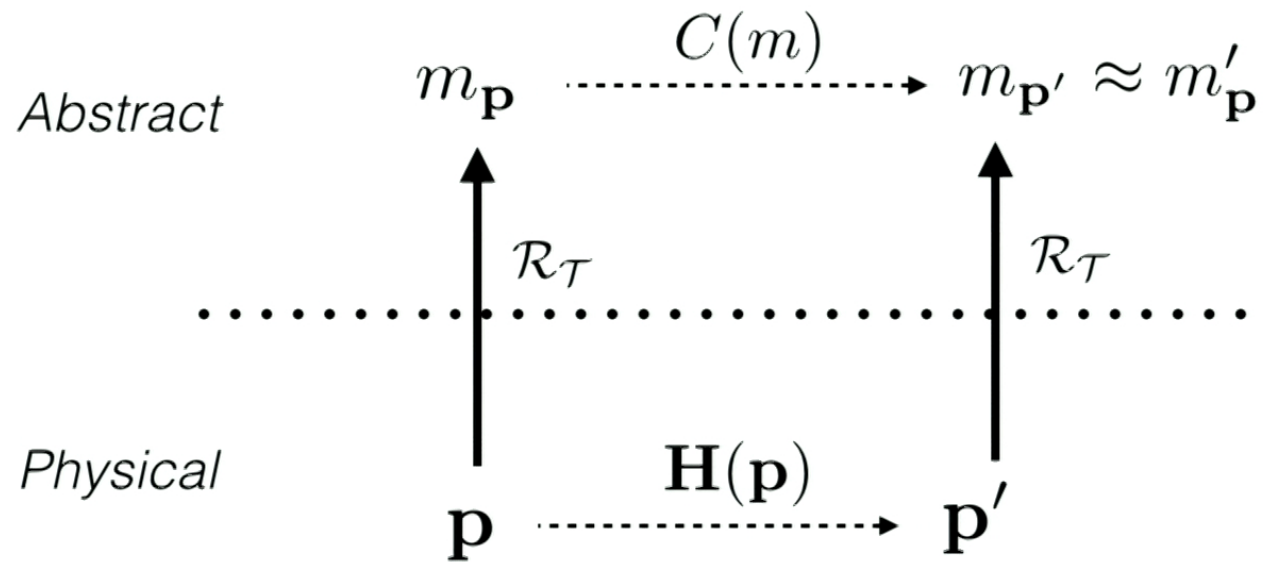
AR Theory

Outcome representation:



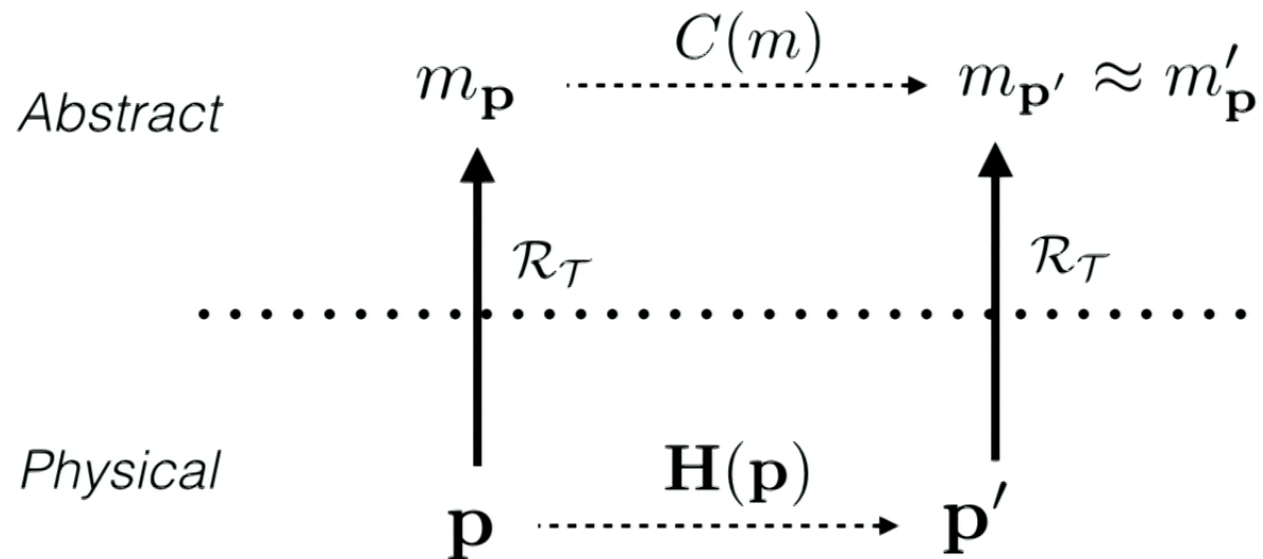
AR Theory

A commuting diagram:



AR Theory

A commuting diagram:



$$\text{iff } |m_{\mathbf{p}'} - m'_{\mathbf{p}'}| \leq \epsilon$$

AR Theory

A minimal requirement of a “good theory” is that it gives commuting diagrams.

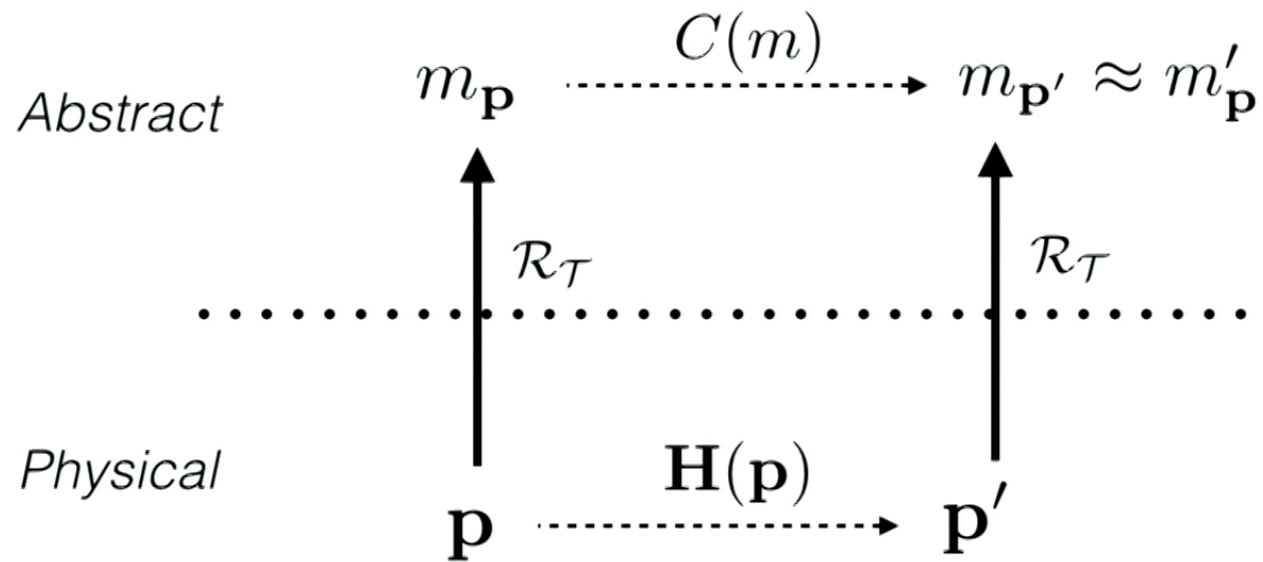
Note: this is not the only requirement

Different theories T give different models for the same physical system.
Testing a theory is testing for a set of commuting diagrams.

What is “good enough” is a matter for the philosophy of science.

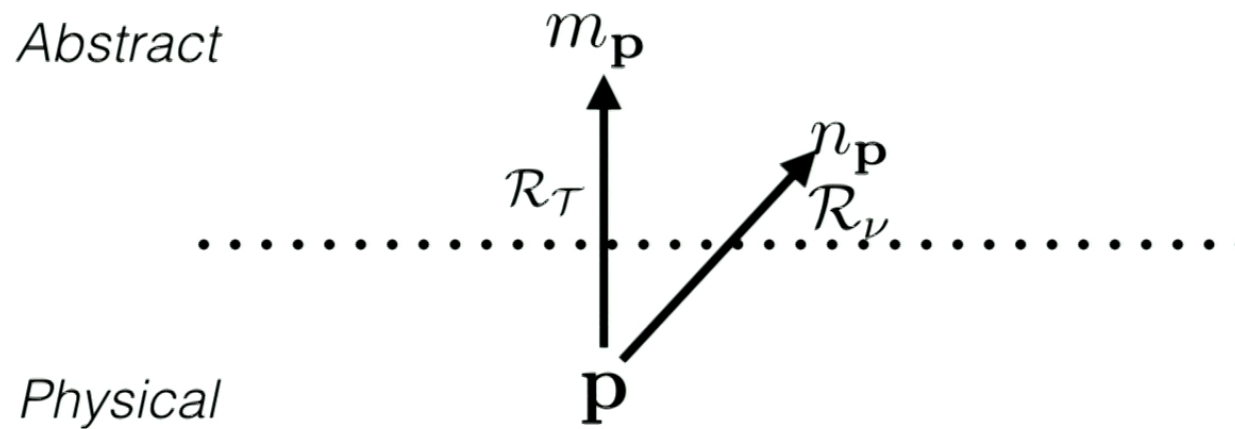
AR Theory

A commuting diagram:



No fundamental representation

AR theory allows for multiple representations of the same system.



No fundamental representation

An example of a hierarchy as often seen:

Psychology → Biology → Chemistry → Fundamental Physics

No fundamental representation

An example of a hierarchy as often seen:

Psychology → Biology → Chemistry → Fundamental Physics

WHAT IS THIS?

Is it quantum mechanics? Quantum field theory? String theory? (Quantum) information theory?

No fundamental representation

An example of a hierarchy as often seen:

Psychology → Biology → Chemistry → Fundamental Physics

WHAT IS THIS?

Is it quantum mechanics? Quantum field theory? String theory? (Quantum) information theory?

Algorithmic information theory?

We should stop basing our metaphysics on a “theory of everything” that does not exist.

No fundamental representation

Representation is always representation **within a theory**.

Representing the external world as a bit string (Muller, Hutter) is a representation in a physical theory.

E.g. data gathered by a robot:

What is the camera on the robot?

What level of detail does it see?

Full e.m. spectrum?

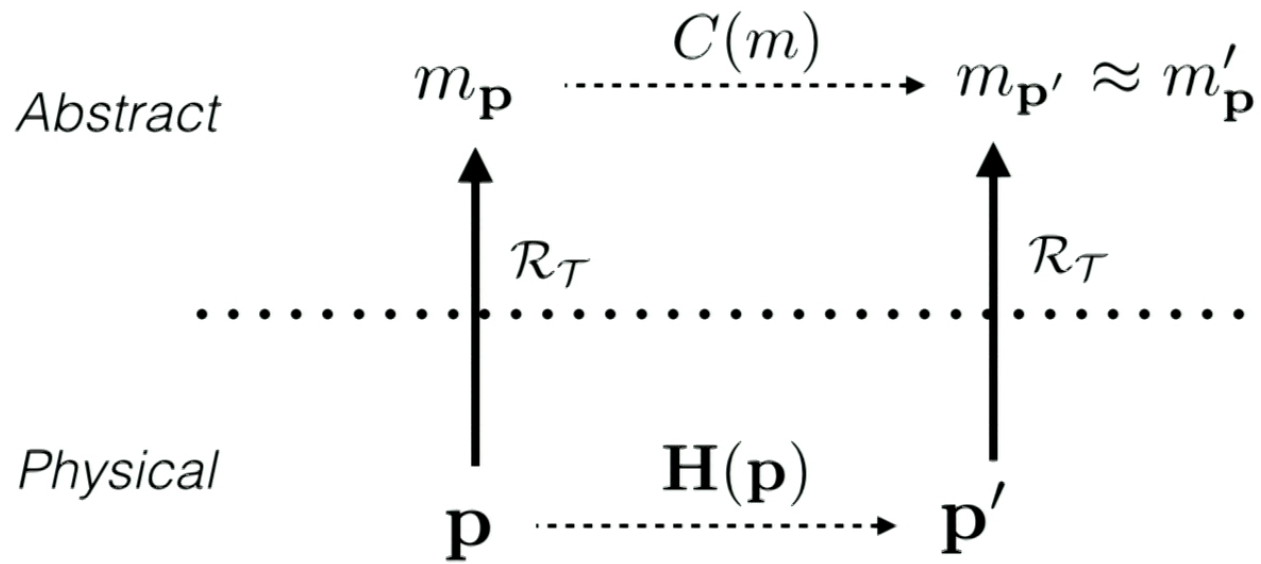
What resolution - optical zoom, electron microscope, gravity wave detector...?

This is what negated Carnap's search for an observer-independent sense-data language: no theory-independent observations. Not even for a robot. (At the very least: I am not hallucinating).

AR theory: no "universal concrete semantics" for the world. Physics is part of the foundations of CS irreducibly. No "computational idealism" for the world.

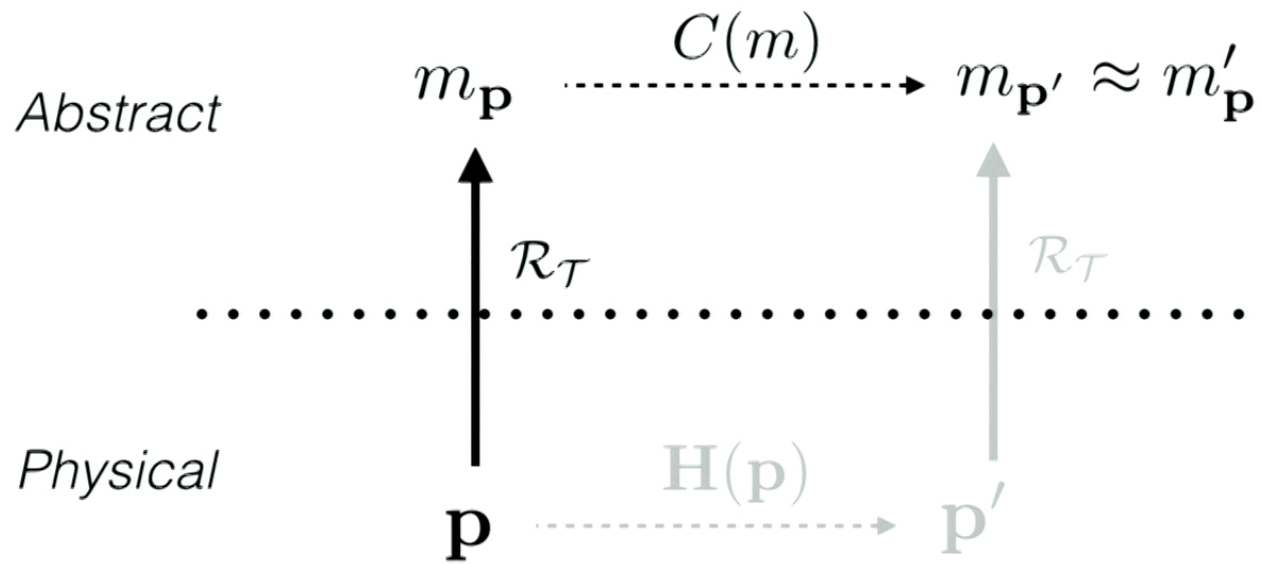
AR Theory

Prediction:



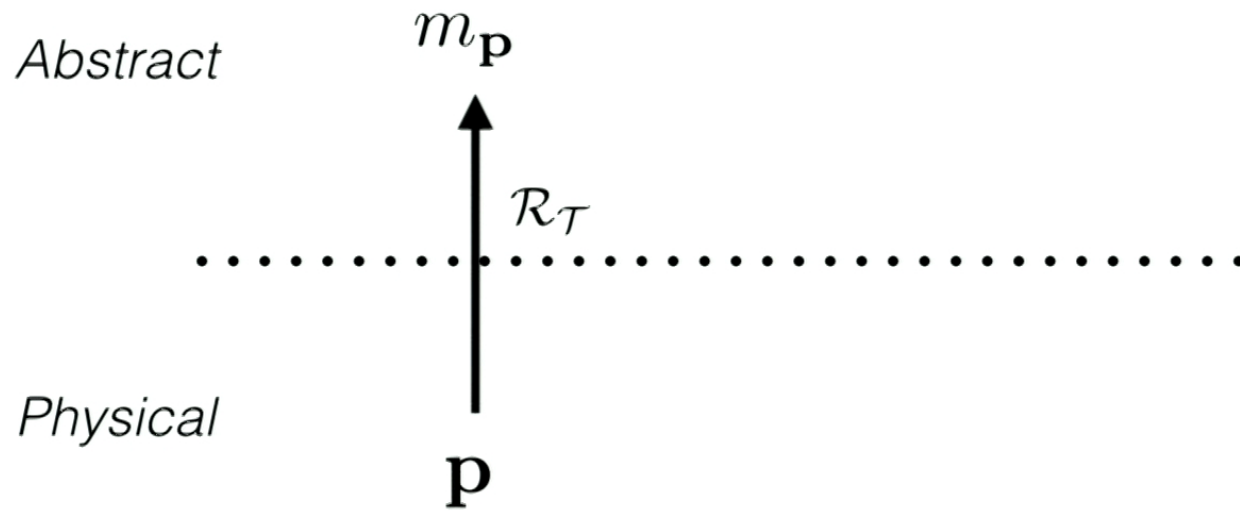
AR Theory

Prediction:



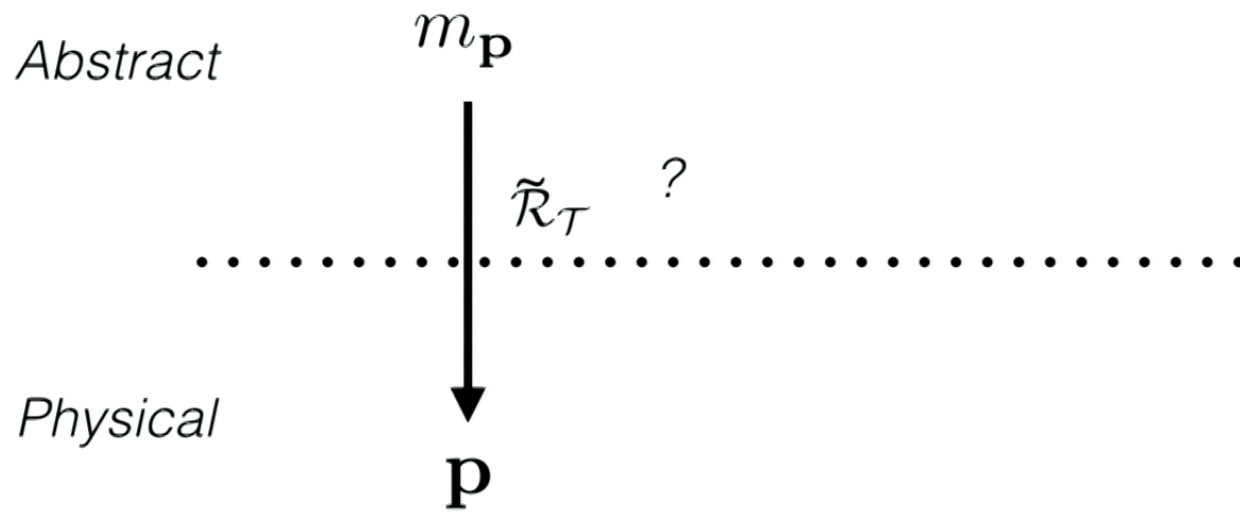
AR Theory

Representation has been given as **directed**. Can it be reversed?



AR Theory

Representation has been given as **directed**. Can it be reversed?



AR Theory

Representation has been given as **directed**. Can it be reversed?

Reversing $\mathcal{R}_{\mathcal{T}}$ depends on how much confidence we have in \mathcal{T} .

If we know the theory well enough to say that it will produce commuting diagrams, we can ask:

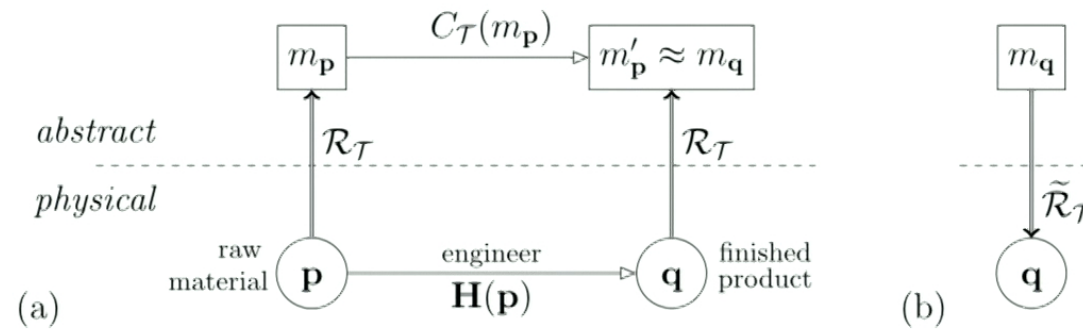
Which set of \mathbf{p} , $\mathbf{H}(\mathbf{p})$, $\mathcal{R}_{\mathcal{T}}$ will give a physical state \mathbf{p}' whose representation is the desired state $m_{\mathbf{p}'}$ to instantiate?

This is engineering.

It requires **skill** and **creativity**: reversing the representation relation is not an algorithmic process.

AR Theory

The instantiation relation and engineering:



An engineered instantiation triple: $\langle m_{\mathbf{p}}, \tilde{\mathcal{R}}_{\mathcal{T}}, \mathbf{P} \rangle$

AR Theory

But what about computers?

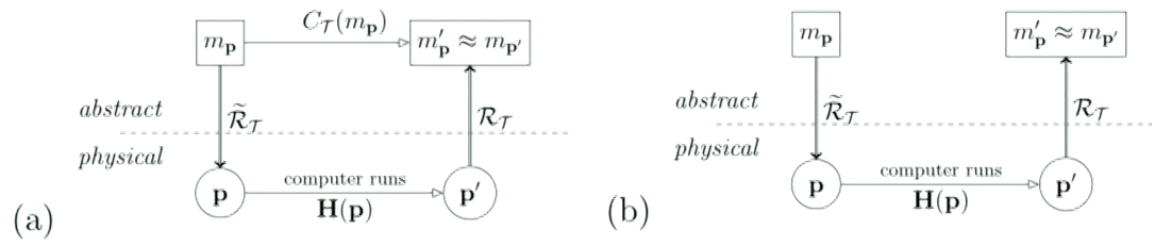
A computer is a device with a well-known \mathcal{T} , sufficient to allow both a modelling and an instantiation relation, $\{\mathcal{R}_{\mathcal{T}}, \tilde{\mathcal{R}}_{\mathcal{T}}\}$

Computing starts with an **abstract** initial state and an **abstract** evolution to run.

The first step is **instantiation**, or, here, **initialisation**.

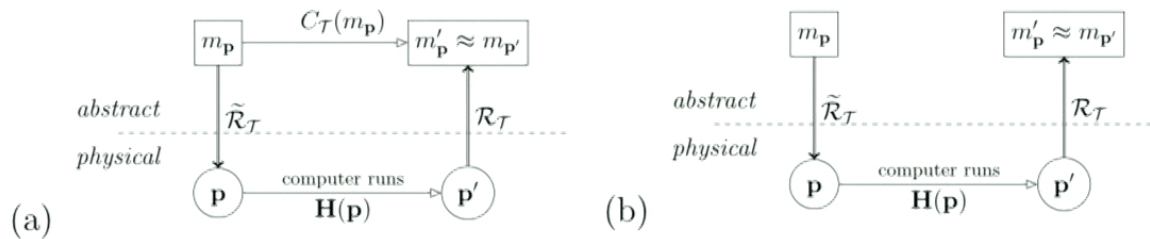
Unless the physical device, the computer, is well-understood, it cannot have a reversed representation relation.

AR Theory



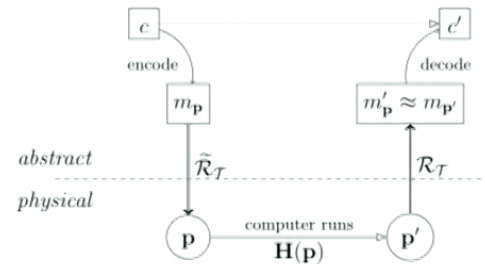
Running a computation on a computer.

AR Theory



Running a computation on a computer.

More precisely: embedding a computation in assembly/machine code and then running on a computer.



**Computing is the use of a
physical system to predict the
outcome of an abstract evolution**

What is computing?

Computing vs engineering/science requires all the following:

A (usually highly-engineered) device in the physical domain with a good and valid theory \mathcal{T}

Representation and instantiation relations $\{\mathcal{R}_{\mathcal{T}}, \tilde{\mathcal{R}}_{\mathcal{T}}\}$

Relevant commuting diagrams over a specific domain of inputs and range of computational operations

A full compute cycle, with the physical device predicting abstract problem-space evolution

Example: rocks don't predict anything, so Putnam's doesn't compute.

“No computation without representation”



Computing example

Classical digital computing:

Theory: The theory of classical computing covers the hardware (including how the transistors implement Boolean logic, and how the architecture implements the von Neumann model) and software (including programming language semantics, refinement, compilers, testing and debugging).

Encode: The problem is encoded as a computational problem by making design decisions and casting it in an appropriate formal representation.

Instantiate: Instantiation covers the hardware (building the physical computer) and software (downloading the program and instantiating it with input data).

Run: The program executes on the physical hardware: the laws of physics describe how the transistors, exquisitely arranged as processing units and memory, and instantiated into a particular initial state, act to produce the systems final state when execution halts.

Represent: The final state of the physical system is represented as the abstract result, for example, as the relevant numbers or characters.

Decode: The represented computational result is decoded into the problem's answer.

Refinement

Relation of ART to standard CS: putting in lower-level physical foundations underneath a typical refinement stack.

This is explicitly the lack of a universal concrete semantics/machine code/assembly language: all in the theory of the physical system, and rooted in the physical system itself not its abstract representation —

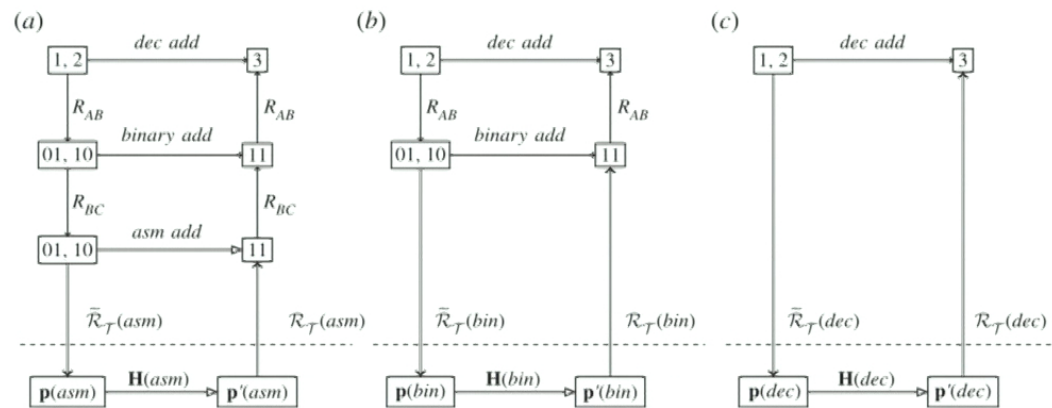


Figure 7. Physical computation, with layers of refinement R on top for base ten (decimal) addition (*dec add*), binary addition (*binary add*) and assembly language addition (*asm add*). Note the physical device and representation differ in each case.

Representational entities

A representational entity grounds the representation relation.

It is important not to **smuggle ourselves in** as representational entities without taking note.

Key distinction: “we can use the system to process information” vs. “the system is processing information intrinsically”.

The difference is whether the representational entity is internal or external to the system. The systems are said to be **closed under representation** or **open under representation**.

Do we represent the system as containing representation, or is the only representation present coming from us *qua* (computer) scientists?

Representational entities

But how do we tell whether representation is occurring in the systems?

We interrogate our best physical theories of the system

But note: the question “is representation occurring” is relative to a theory of the system. Different levels of description may (will) return different answers.

The question is then, for a theory T, is there a valid semantics of the system processes that includes *within itself* a description of (parts of) the system using compute cycles and representation?

Representational entities

We are looking for **encoding and decoding**, and **semantics apart from physics**.

How do we argue for the latter? A hallmark of representation is **arbitrariness of the encoding**: same semantics, different language.

We are looking for highly-engineered systems.

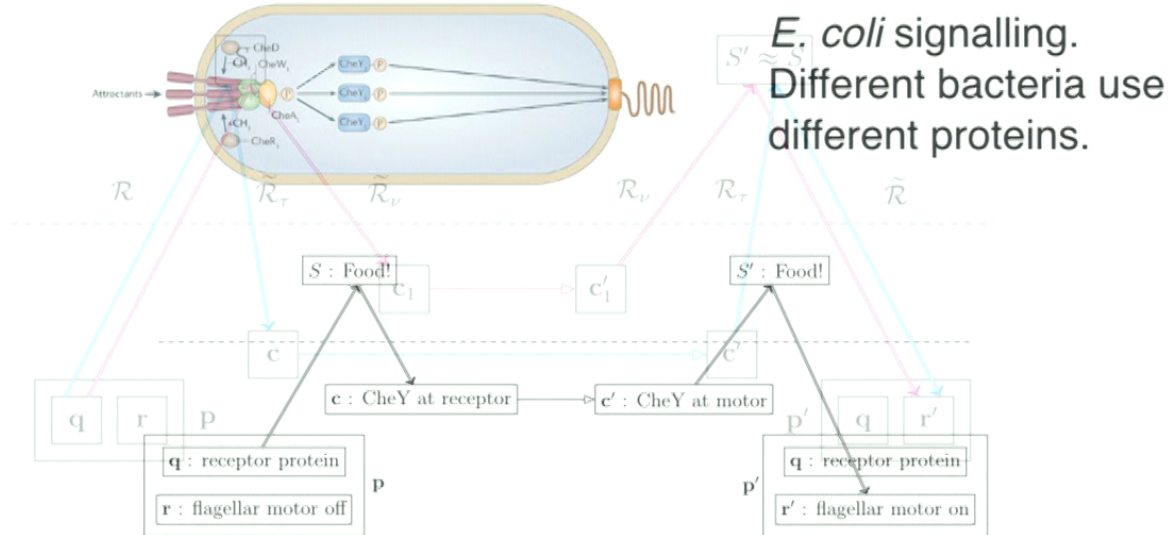
In biology, we can see the process of engineering and different outcomes that produce the same representational semantics.

Evolution is engineering biology.

Bacteria

Bacterial chemotaxis is an instance of signalling.

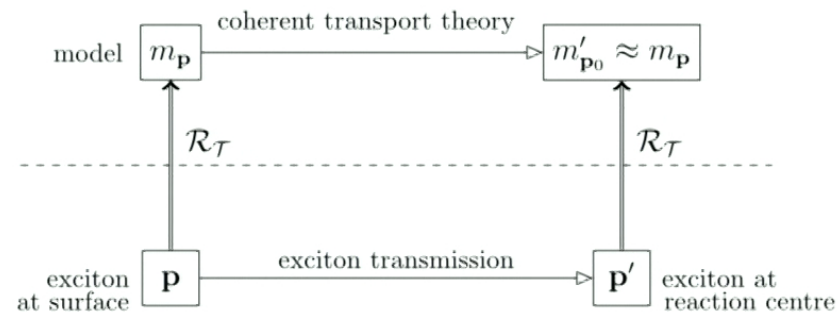
Same signal, different pathways:



Photosynthesis

Light-harvesting complexes transmit photons to the reaction centre via coherent (quantum) energy transport.

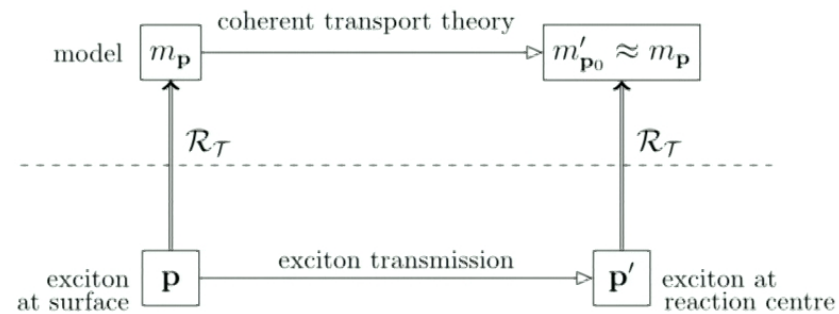
This is the same mechanism as used in quantum walk computation.



Photosynthesis

Light-harvesting complexes transmit photons to the reaction centre via coherent (quantum) energy transport.

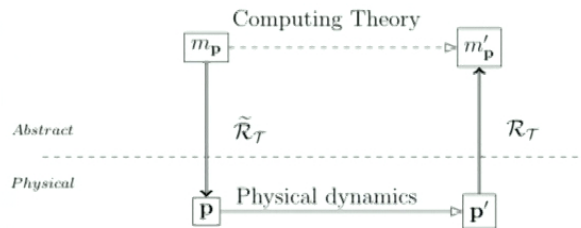
This is the same mechanism as used in quantum walk computation.



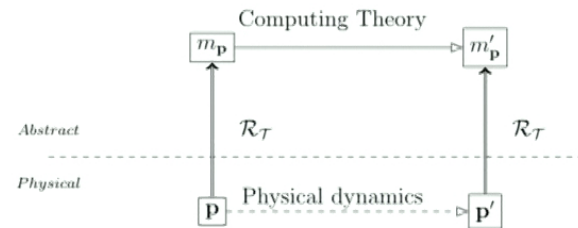
But this is **science**: no representational step within the system itself

Intrinsic logics

A computational description of a system can take part in two different AR cycles:



1. Compute cycle



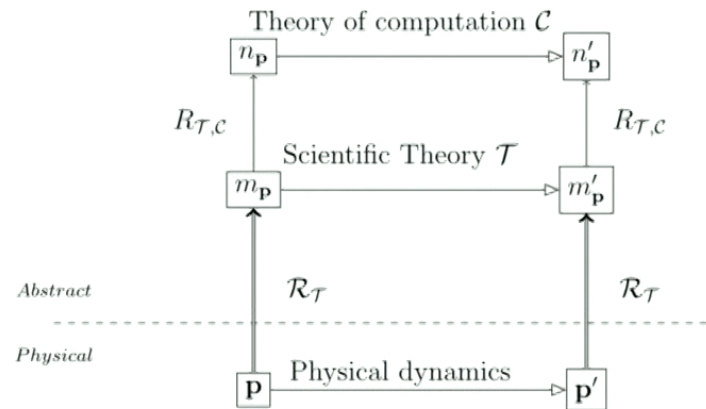
2. Computational description

Cycle 2. is used in developing a computer.

It can **also** be used to describe the dynamics of a physical system.

Intrinsic logics

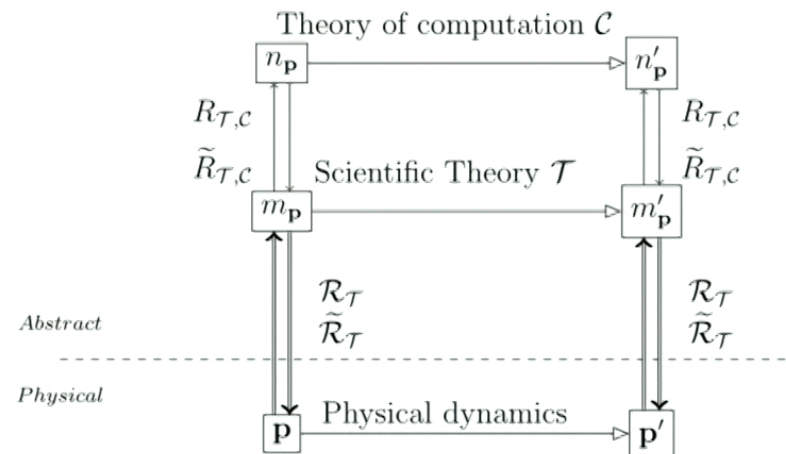
The computational theory is not basic: it requires a physics/chemistry/etc to base on (eg. to define available degrees of freedom).



Intrinsic logics

Once a computational theory is well developed, along with the scientific theory, they can both be used to instantiate.

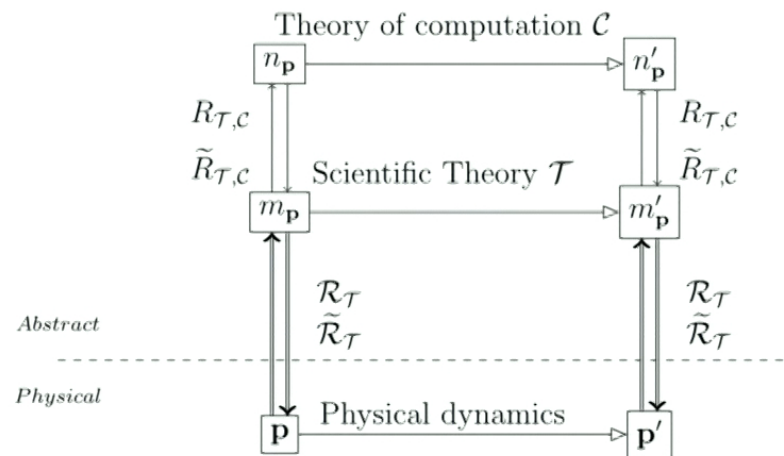
This is the desired outcome of a well-characterised novel substrate:



Intrinsic logics

Different substrates support different computational models.
So: use the substrate to inform the logic, based on experiments.

Don't impose a logic on the system: use what the substrate is good at to construct unconventional program logic for unconventional systems.



Quantum computing

Can quantum computers compute some things faster than classical computers?

If so, what problems can they do this for?

Why?

Quantum computing

Can quantum computers compute some things faster than classical computers?

If so, what problems can they do this for?

Why?

What does an answer to these questions look like?

Examples:

Quantum computers store and process many bits of data at once

Quantum algorithms weave patterns of amplitudes

Quantum computing is like classical probabilistic computing with negative probabilities

None are particularly appealing. What is the true answer?

Quantum computing

My answer:

We don't know.

Quantum computing

My answer:

We don't know.

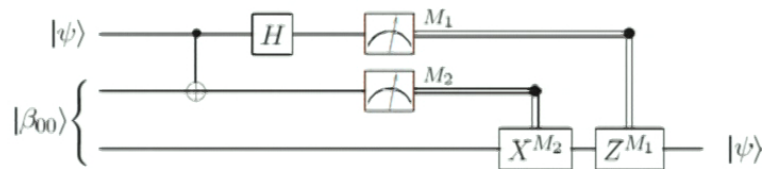
And we don't know because we don't have a solution to the problem of interpretation of QM.

Quantum computing

To give an explanation about quantum processes is to give a description in a higher-level language.

Quantum computing **is** a story about quantum processes (cf textbooks).

However — so far we're stuck at the lowest level of refinement (CNOT etc):



```
import pyquil.quil as pq
from pyquil.gates import *
import pyquil.api as api

qvm = api.SyncConnection()

ins = pq.Program()

ins.inst(H(1), CNOT(1,2)) #Creating B00
ins.inst(H(0), Z(0), CNOT(0,1), H(0))
ins.measure(0,0).measure(1,1).if_then(1, X(2)).if_then(0, Z(2))

wvf, addr = qvm.wavefunction(ins, [0,1])
print addr, wvf
```

Teleportation: circuit and Pyquil (Rigetti)

<https://medium.com/@rasa97uchiha/hello-world-s-quantum-teleportation-using-rigettis-forest-64b9afda7ba>

Quantum computing

How do we go up the refinement stack? By finding higher-level processing concepts for QC (e.g. “quantum queueing”).

Novel HLLs for quantum computing will be novel HLLs for fundamental quantum processes.

“Shut up and calculate” is no good for a high-level intrinsic logic. We need a human-readable explanation.

The problem of generating high-level languages for quantum computers is now identical with the problem of developing realist interpretations/causal models for quantum theory.

Conclusions

Computing is physical. To use the language of computation in physics, need to keep the physical system part of the framework.

The physical world is not isomorphic with a computational representation of it. The world is not a bit string.

Put physics into the foundations of CS — AR theory does not reduce/replace physics with computational arguments.

Computing is the use of physical system to predict abstract computational evolution.

For non-standard computing systems, the challenge is to produce intrinsic computational logics.

These intrinsic logics can then also be used as high-level languages for the physics of the system.

Quantum computing is the most immediate application for HLLs.