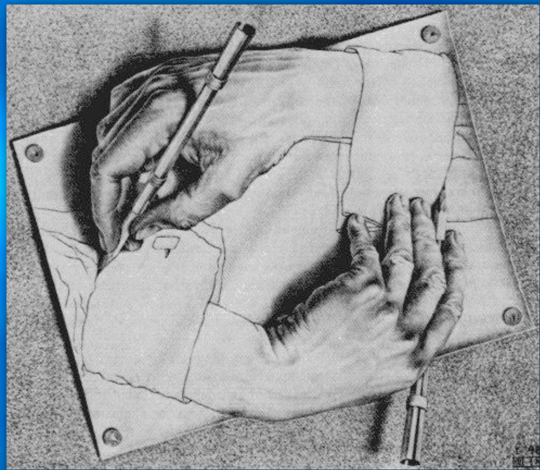
Towards Autopoietic Computing

Gerard Briscoe and Paolo Dini
Digital Ecosystems Lab
Department of Media and Communications
London School of Economics and Political Science

Autopoiesis



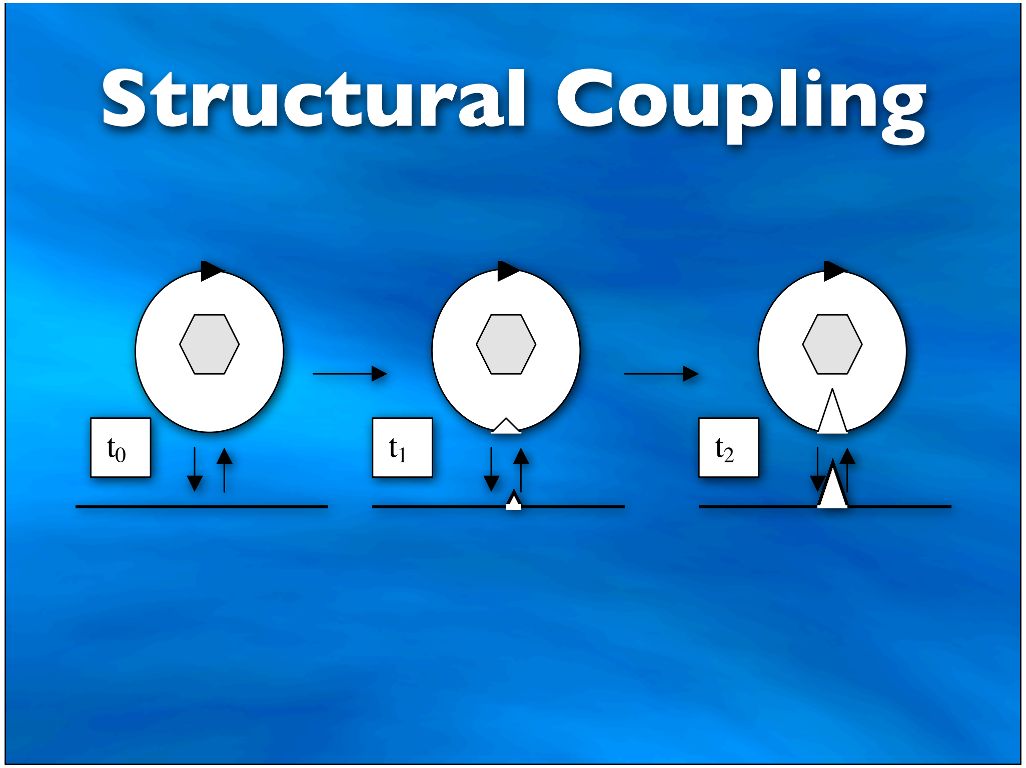
Autopoiesis literally means "auto (self)-creation", and expresses a fundamental dialectic between structure and function.

The term was originally introduced by Chilean biologists Humberto Maturana and Francisco Varela in 1972:

An autopoietic machine is a machine organized (defined as a unity) as a network of processes of production (transformation and destruction) of components which: (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in space in which they (the components) exist by specifying the topological domain of its realization as such a network. [1]

"A good artistic visual expression of autopoiesis is Escher's "Drawing Hands," reproduced here. The two complementary hands can draw each other, but one hand cannot draw itself." – John David Garcia [3]

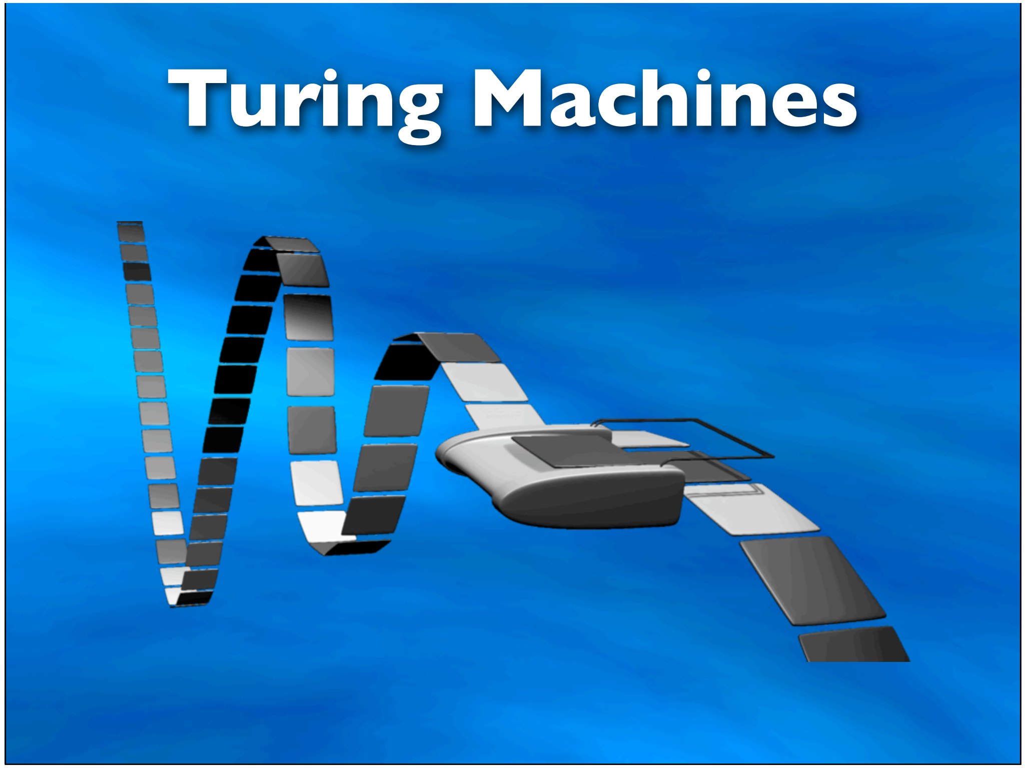
The term autopoiesis was originally presented as a system description that was said to define and explain the nature of living systems. A canonical example of an autopoietic system is the biological cell. The eukaryotic cell, for example, is made of various biochemical components such as nucleic acids and proteins, and is organized into bounded structures such as the cell nucleus, various organelles, a cell membrane and cytoskeleton. These structures, based on an external flow of molecules and energy, produce the components which, in turn, continue to maintain the organized bounded structure that gives rise to these components.



Structural Coupling – The autopoietic system, represented by a circle, defined by its structure and its organisation, initially confronts a medium without organised objects.

As recurrent interactions between the medium and the system are stabilised, at t_1, an object begins to be configured.

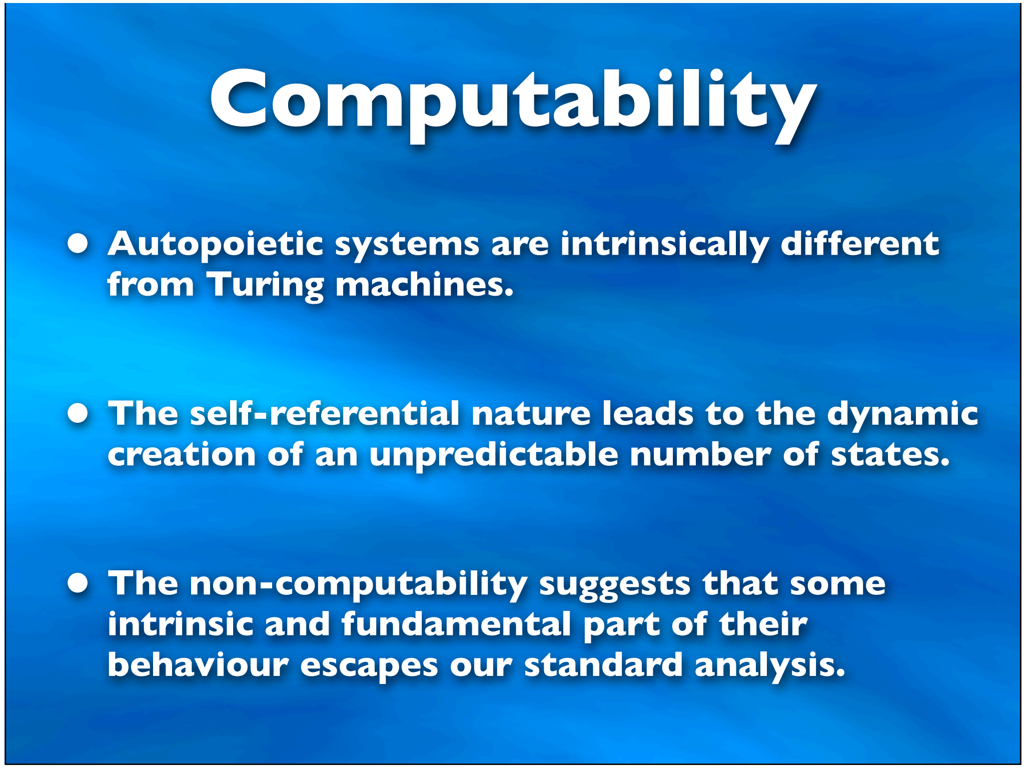
The object is made of two complementary parts. One part exists in the medium, and the other exists as a change in the autopoietic system structure.



Turing Machine - A basic abstract symbol-manipulating device which, despite its simplicity, can be adapted to simulate the logic of any computer algorithm.

The Turing Machine mathematically models a machine that mechanically operates on a tape for which symbols are written which it can read and write one at a time using a tape head.

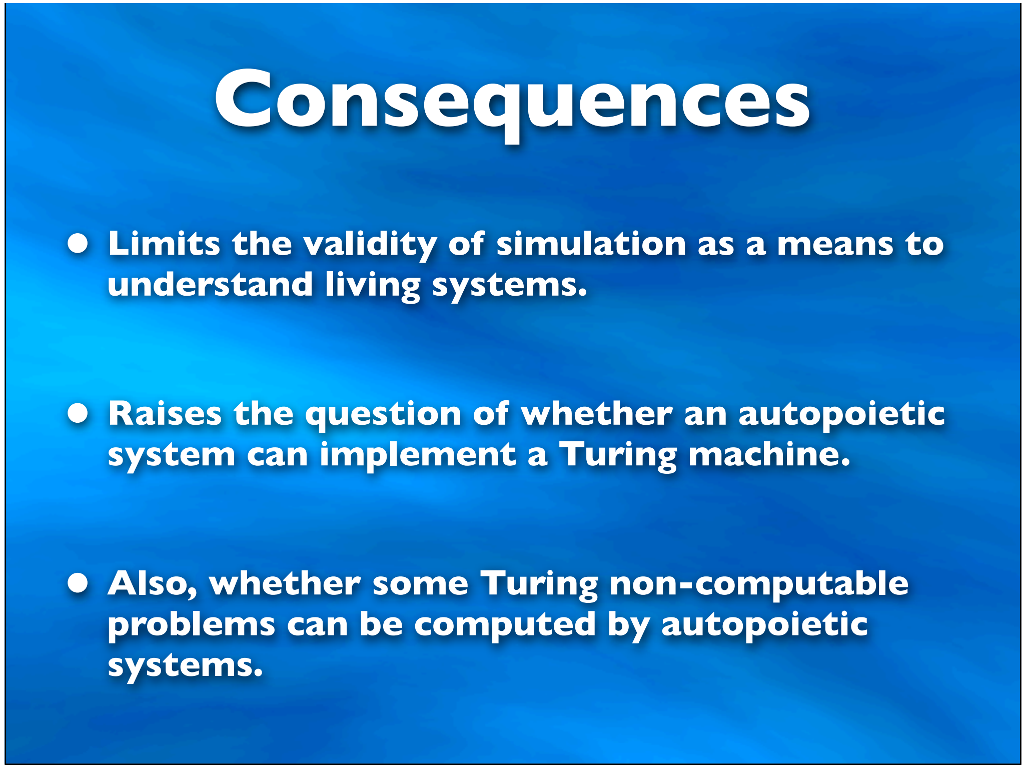
Operation is fully determined by a finite set of pre-determined elementary instructions contained within the Turing machine.



Autopoietic systems are intrinsically different from Turing machines... They cannot be simulated by Turing machines as they are not Turing-computable.

The self-referential nature of circularity that characterises autopoietic systems leads to the dynamic creation of an unpredictable number of states.
The dynamic creation of an unpredictable number of new states implies that no upper bound can be placed on the number of states required.
While the Church definition of computability assumes that the basic operations of a system must be finite.

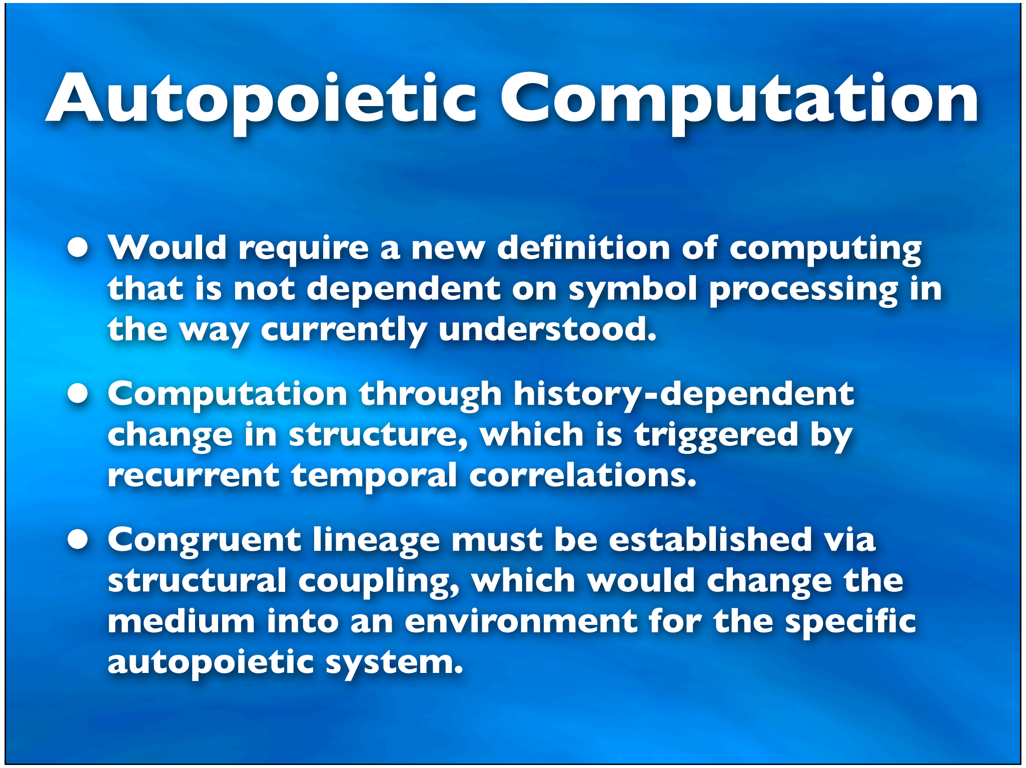
The non-computability of autopoietic systems suggests that some intrinsic and fundamental part of their behaviour escapes our standard analysis based on phase states and/or evolution equations.



Limits the validity of simulation as a means to understand living systems, showing that the phenomenology that arises from the circularity of metabolism cannot be simulated with current computer architectures, those based on the Von-Neumann implementation of Turing machines.

The inapplicability of the Turing-Church thesis for autopoietic systems also opens some important new questions. The first is to analyse whether an autopoietic system can implement a Turing machine.

Also, whether some Turing non-computable problems, like the busy beaver, can be computed by autopoietic systems.



Would require a new definition of computing that is not dependent on symbol processing, because the unfamiliar computing aspects of autopoietic systems arise primarily from the internal reference frame of the controller. The control is through the logic (which could perhaps be modelled as symbols) of the maintenance of circular organisation in the presence of structural coupling.

Computation through history-dependent change in structure, which is triggered by recurrent temporal correlations. This change is the consequence of the recurrent interactions between the autopoietic system and its medium.

An autopoietic system must be introduced in such a medium and a congruent lineage must be established via structural coupling, which would change the medium into an environment for this specific autopoietic system.

Automaton

C = (A, B, Q, λ,δ)

A = input alphabet
B = output alphabet
Q = state set
λ = state transition function
δ = output function

A & B are finite but Q is not necessarily finite

InteractING Machine

Sequential machine:

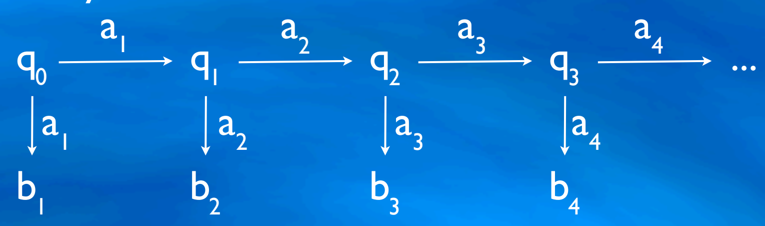
Definition 3.1. Let A be a non-empty set. Then $A^+ = \{(a_1, \ldots, a_n) : n \geq 1, a_j \in A\}$. A *sequential machine* is a function $f : A^+ \to B$, where A is the basic input set, B is the basic output set, and $f(a_1, \ldots, a_n) = b_n$ is the output at time n if a_j is the input at time j for $1 \leq j \leq n$.

Generalisation: interactING machine:

Definition 3.2. Let $f : A^+ \to B$ be a sequential machine. Then an *interacting machine* $f^+ : A^+ \to B^+$ is defined by

$$f^+(a_1, \ldots, a_n) = (f(a_1), f(a_1, a_2), \ldots, f(a_1, \ldots, a_n)).$$

Realised by 'circuit' or automaton:



On the Realisation of Computability

"The reason why Turing machine programs to realize a computable f are not unique and the circuit which realizes the (sequential) machine f (namely C(f)) is unique is not hard to fathom. In the sequential machine model we are given much more information. It is `on-line' computing; we are told what is to happen at each unit of time. The Turing machine program is `off-line' computing; it just has to get the correct answer -- there is no time restraint, no space restraint, etc." (Rhodes, 2010)

Finiteness of State Set

Rather than constructing dynamical behaviour through a sequential algorithm expressed in a programming language, which can be realised by a single automaton or Turing machine, we are talking about constructing dynamical behaviour through the interaction of two or more *finite*-state automata.

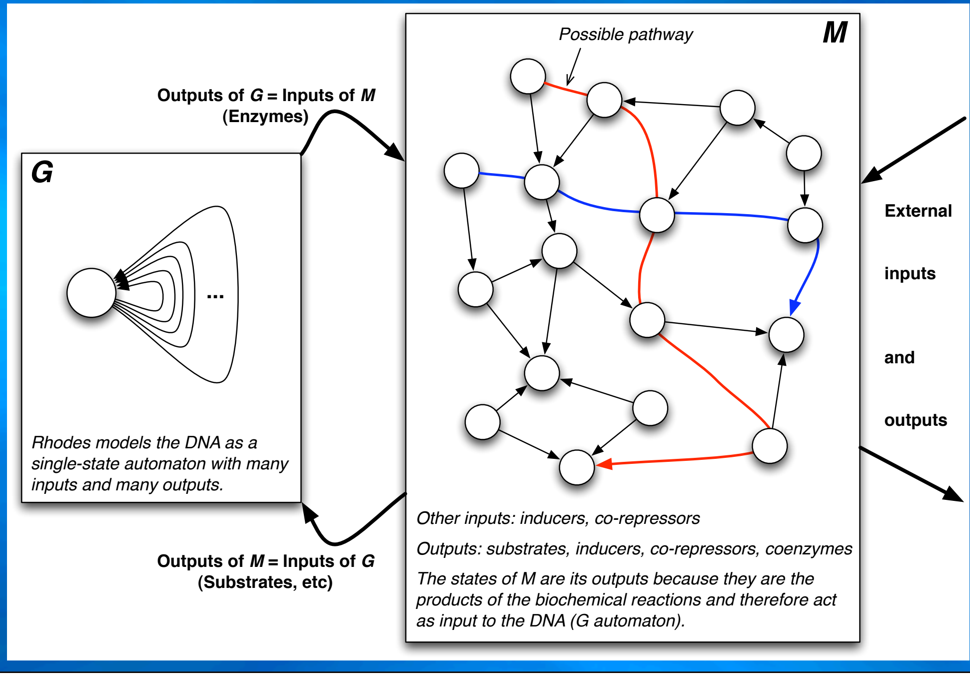
This is because, if the possibility that Q be infinite is left open as the above definition does, "then the output function could be a badly non-computable function, with all the interesting things taking place in the output map δ, and we are back to recursive function theory" (Rhodes, 2010)

InteractION Machine

Therefore, we note the interesting conclusion that, for a tractable approach, Q must be finite and that the realisation of an Interaction Machine must be made up of interacting finite-state automata. Thus, from the mathematical or behavioural perspective, we will build the Interaction Machine by using sequential (or interacting) machines as basic units and by combining them in various ways.

So, an interactION machine is the combination of two or more interactING machines: outputs of one are the inputs of one or more other machines.

Cell as Interaction Machine



Questions?