Coordinate Systems of Understanding

Attila Egri-Nagy
www.egri-nagy.hu

Biological & Neural Computation Laboratory
Science & Technology Research Institute
University of Hertfordshire
United Kingdom

Very Shortly

For any finite system a working hierarchical model can be generated automatically.

Basic Notions – Philosophical Introduction

What is a Model?

Any system, which is not the studied system itself, but has all the basic characteristics of the phenomenon.

It is expected to be simpler in a sense. (1:1 scale are maps are not really useful)

Working Model?

We would like to model the dynamical processes as well, not just the static structure.

(the object-oriented way of thinking)

Disassembling

Mostly, scientific understanding proceeds by taking apart things, identifying their components.

But the list of the ingredients is not a complete recipe.
How do we put together the components?
Our answer here: hierarchically.

Hierarchy – Ethimology

Date: 14th century

1. a division of angels
2. a ruling body of clergy organized into orders or ranks each subordinate to the one above it; especially : the bishops of a province or nation b : church government by a hierarchy
3. a body of persons in authority
4. the classification of a group of people according to ability or to economic, social, or professional standing; also : the group so classified
5. a graded or ranked series <Christian hierarchy of values> <a machine's hierarchy of responses>

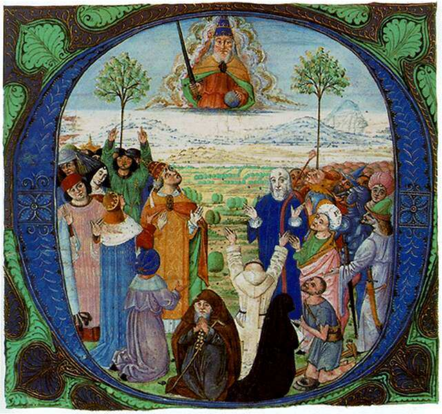
Merriam-Webster

More Recent Definitions

- “hierarchy is a partial ordering — specifically a tree”
- “hierarchy is associated with a very fundamental form of parsimony of interactions. ... discovering the points of cleavage at which the least information needs to be passed”, subroutines in a program
- “Everything is connected, but some things are more connected than the others.”

Herbert Simon

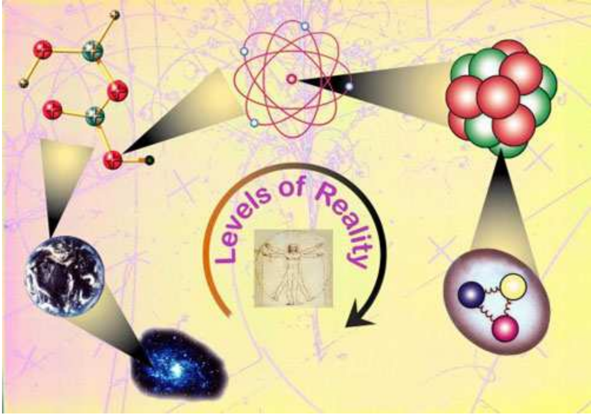
Social Hierarchy - from the Pope to a Beggar



mek.oszk.hu

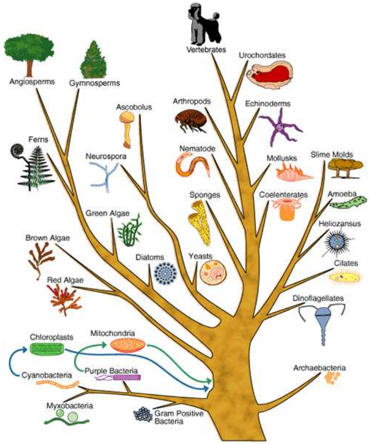
From Quarks to Galaxies

Spatial and containment hierarchies



www.phinuphy.org

Tree of Life

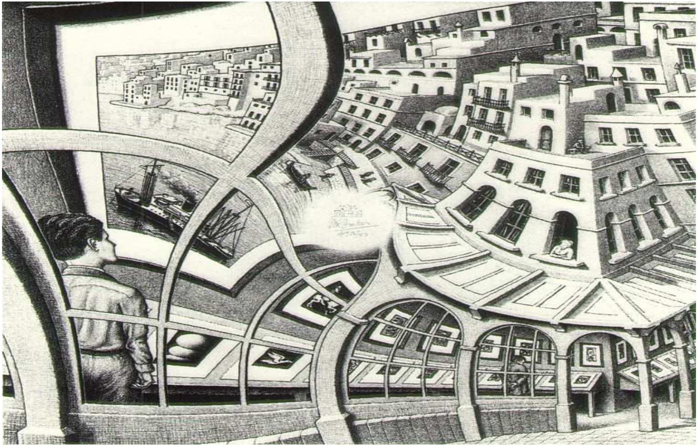


genetics.nbii.org

Nice Properties of Hierarchy

- information flow between levels are restricted enabling modularity (also within one level with parallel components).
- generalization and specialization are natural operations realized by taking subsets of levels in either direction up or down the hierarchy.

In or Out?



M.C. Escher: *Print Gallery* 1956.

Inside or Outside? Philosophically...

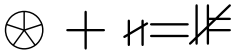
Is hierarchy an inherent property of the world around us, or "only" our own – albeit very efficient – cognitive tool?

We do not really care...

Inside is Enough! e.g.

Decimal notation for integer numbers is an extremely useful tool. The representation supports simple algorithms.

without it we would have to memorize an infinitely large lookup-table of symbols



instead of

$$147 + 532 = 679$$

Coordinate System

By a coordinate system we mean a notational system (in the broadest possible sense), with which we can address the components and their relations in a decomposition, thus gaining a convenient way for grasping/talking about the structure of the original phenomenon.

e.g.

- Non-hierarchical: Descartes coordinates for the Euclidean space
- Hierarchical: our decimal number notation system

Ultimate Goal

For any finite system a working hierarchical model, a coordinate system for understanding can be generated *automatically*.

And by unveiling the chains of the dependencies within the system we can measure the complexity of the system under study.

Algebraic Automata Theory – Decomposition of Computational Structures

Prime Decomposition

The quickest way to explain Krohn-Rhodes Theory is using the metaphor of Prime Decomposition of Integers.

	Integers	**Automata**
Factors	Primes	Flip-flop Automaton Permutation Automata
Composition	Multiplication	Wreath Product
Precision	Equality	Division, Emulation

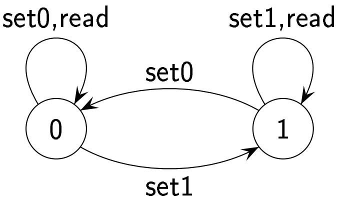
Automata

Grasping the ide of state transitions mathematically. Very strong abstraction focusing on the important notion of change.

- A state set
- X input symbols
- $\delta : A \times X \rightarrow A$ state transition function

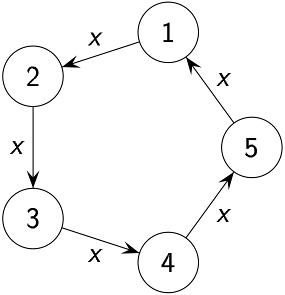
Flip-flop automaton

It is capable of storing one bit information.



Counter

States: $A = \{1, 2, 3, 4, 5\}$
Input symbols: $X = \{x\}$



Types of Computational Processes

- Irreversible, e.g. overwriting a segment of memory
- Reversible

The distinction is more immediate algebraically.

Group Theory

The group is the abstract algebraic way of thinking about symmetries. Operations can be undone.

G set with an associative binary operation $\cdot : G \times G \rightarrow G$, with identity element $1 \cdot g = g \cdot 1 = g$, and with inverses to each elemenet $g \cdot g^{-1} = g^{-1} \cdot g = 1$.

Associativity:

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Semigroup Theory

Generalization of groups. There are irreversible operations (collapsing states).

S set with an associative binary operation,

- There are extremely many: 1.8 billion non-equivalent semigroup, $|S| = 8$.
- Less is more: every group is a semigroup.
- They represent finite computations.
- Due to associativity it can be considered as a model of time.

Semigroup Examples

- $(\mathbb{N}, +)$, 0 is the identity
- $(\mathbb{Z}, +)$, 0 is the identity, there exist inverse elements $n + (-n) = 0$, therefore it is a group
- $(\mathbb{N}, \cdot)$, is the identity
- $(\{a, b\}, \circ)$

$\circ$	**a**	**b**
a	a	b
b	a	b

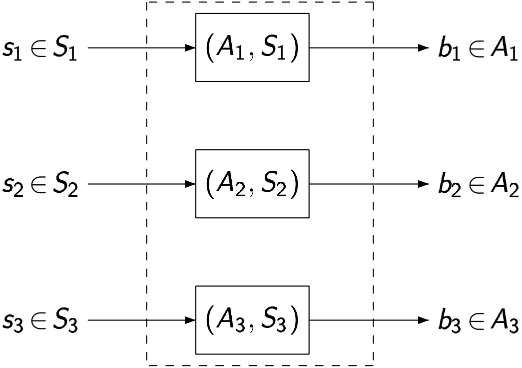
Transformation Semigroup

How can we make automata from semigroups? (A, S)
$A = \{1, 2, 3, 4, 5\}, S = \{x, x^2, x^3, x^4, x^5 = 1\}$

$$x = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 1 & 2 \end{pmatrix}$$

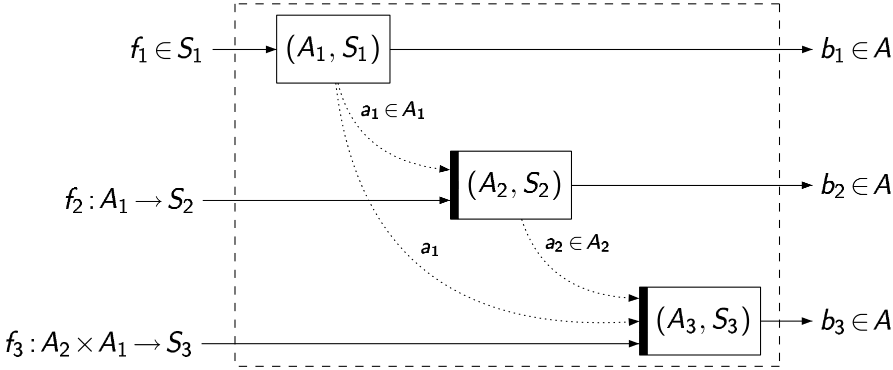
Parallel composition



$(A_3, S_3) \times (A_2, S_2) \times (A_1, S_1)$

$(a_3, a_2, a_1) \cdot (s_3, s_2, s_1) = (b_3, b_2, b_1) = (a_3 \cdot s_3, a_2 \cdot s_2, a_1 \cdot s_1).$

Cascade Composition (Wreath Product)



$(A_3, S_3) \wr (A_2, S_2) \wr (A_1, S_1)$

$(a_3, a_2, a_1) \cdot (f_3, f_2, f_1) = (b_3, b_2, b_1) = (a_3 \cdot f_3(a_2, a_1), a_2 \cdot f_2(a_1), a_1 \cdot f_1)$

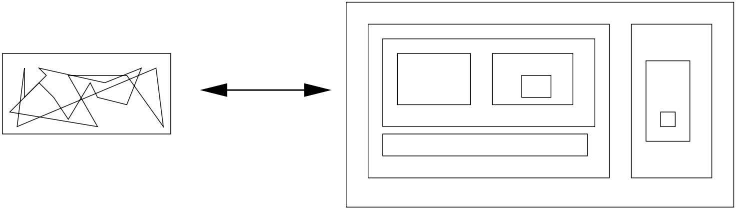
Example: Bidirectional Counter

Coordinates: (n, mode)
Modes: adding $+$, subtracting -
Two operations: counting and switching mode.

Hierarchical dependence: the counting operation does different things depending on the top level coordinate.

Emulation

(B, T) emulates (A, S)-t, if any calculation in (A, S) can be done in (B, T). (Using algebraic terminology: (A, S) divides (B, T))



(A, S) $\qquad$ $(B_1, T_1) \wr \ldots \wr (B_n, T_n)$

Any finite state automata $\mathcal{A}$ can be emulated by a cascade product of simpler automata. The components are the flip-flop and those group automata, that $\mathcal{A}$ is able to emulate.(And the reverse direction: if it can be built from certain irreducible automata, than $\mathcal{A}$ can emulate those components.)

Computational Implementations

constructive proof $\simeq$ algorithm

- $V \cup T$ (Krohn, Rhodes, 1962)
- Holonomy (Zeiger 1968, Eilenberg 1976)
- Categories (Rhodes, Weil, 1989)
- Semigroup expansions (Rhodes 1991)
- $\mathcal{L}^+$ decomposition (Nehaniv, 1996)
- Kernels (Ésik 2000)
- Holonomy (Elston, Nehaniv 2002)

But there was no computational implementation before this work.

Why?!?

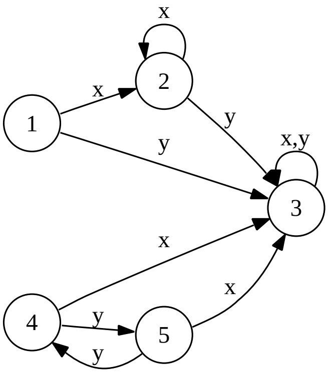
- The difficulties are in mathematics, but the motivation comes from somewhere else. Interdisciplinary approach is needed.
- At the time of the birth of the theory the available computational power was not enough.
- It is computationally challenging to calculate with semigroups (there are too many elements).

Beware of Combinatorial Explosions!!!

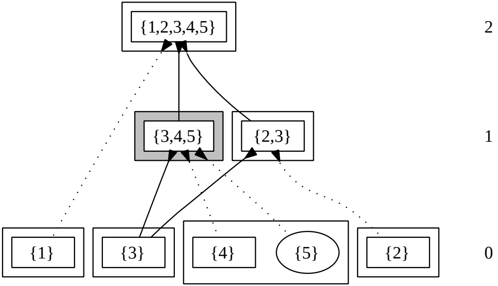
Incremental Version

Currently we are working on the incremental version of the decomposition algorithm : just calculating some of the upper levels. This way we get finer and finer approximations buy using more computation.

Example: randomly generated automaton...

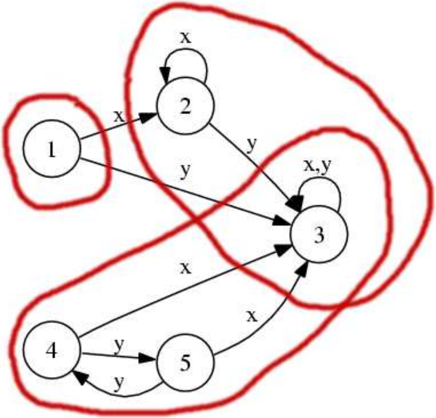


...and its holonomy decomposition

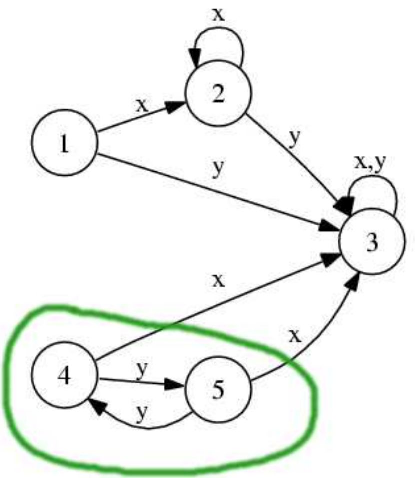


$$(\mathbf{2}, \overline{1}) \times (\mathbf{2}, \overline{C_2}) \wr (\mathbf{3}, \overline{1})$$

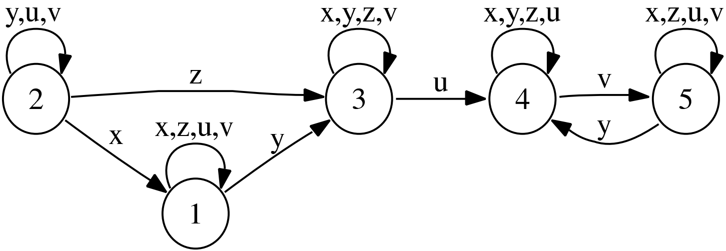
…giving an approximation reset automaton



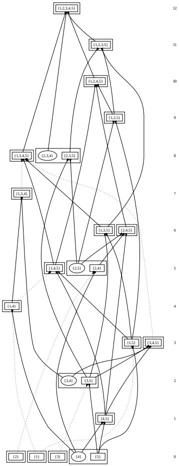
…identifying the reversible computation



An evolved automaton...



...and its holonomy decomposition



Applications – Promises and Difficulties

An Application in Math

Decomposing finite residue class rings of integers modulo n yields a nice structural theorem for their holonomy decomposition.

It is interesting from the computer algebraic point of view, since the decompositions give us coordinatizations which are closely related to, but also unlike, our usual base 10 system, and might be well-suited for applications in computer arithmetic.

Measuring Complexity

The number of hierarchical levels show the length of the maximal functional dependency chain within the original system. (Group complexity: the number of alternations between group and flip-flop components.)
This measure can be axiomatized (Rhodes, Nehaniv, 1996.)

- Basic building blocks
- Constructibility
- Covering
- Bounded emergence
- Noninteraction

Understanding Complicated Biological Machinery

Getting more than a single-valued measure: a map for structural complexity.
Current research: understanding genetic regulatory networks.

Applying KRT for analysing artificially evolved GRNs. Converting Petri-Net models to finite state automata.

Formal Models of Understanding

The automatically generated coordinate system can serve as the dynamically changing (regenerated) inner representation of the environment for any kind of intelligence.

Though, we still do not really know how to calculate efficiently with the coordinates.

Difficulties

Our method can be applied for promoting understanding whenever a complicated phenomenon can be described as a finite automaton. But,

- How to get a FA description? e.g. ϵ-machine reconstruction from a sequence of observations.
- Isn't it too much abstraction? Maybe we need to reformulate KRT for the probabilistic realm.

Thank You for Your Attention!

More information available at:

`http://graspermachine.sf.net`

REMEMBER!!!
If You have finite state automata then we can tell You how to understand them exactly! *

* Size restrictions may apply.