~~Hierarchical Decomposition: An Intuitive Approach~~
Holonomy Decomposition: Finally Some Explanations

Attila Egri-Nagy   Chrystopher L. Nehaniv

Centre for Research and Mathematics
School of Computing, Engineering and Mathematics
University of Western Sydney

———

Centre for Computer Science and Informatics Research
University of Hertfordshire

2014.06.19.

Big, abstract ideas and how they appear in the holonomy decomposition of finite transformation semigroups.

abstract idea	**holonomy decomposition**
approximation	?
compression	?
hierarchical structure	?

approximation

Using a sequence of simpler systems, in which the members are increasingly similar to but not the same as the original system to be approximated.

- a curve imitated by increasingly more straight line segments with smaller and smaller length

compression

When several copies of the same component exist in a system, then it is enough to deal with a single copy of the components and the locations of the copies.

- units with the same floorplan in a towerhouse
- subgroup and a transversal (coset representatives)

hierarchical structure

Control information goes in one direction only – component connections are defined by a directed acyclic graph, but in practice we do rooted trees.

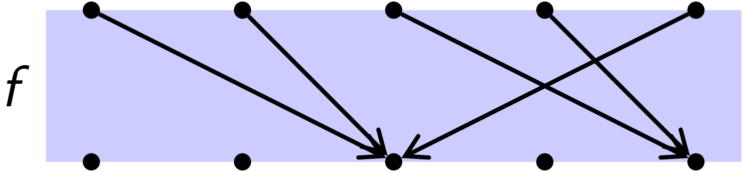
- arithmetic
- robotic arm

statement

In the holonomy decomposition of finite transformation semigroups we *approximate* states and transformations, then we *compress* approximation data and and finally put the encoded form into a hierarchical structure (cascade product).

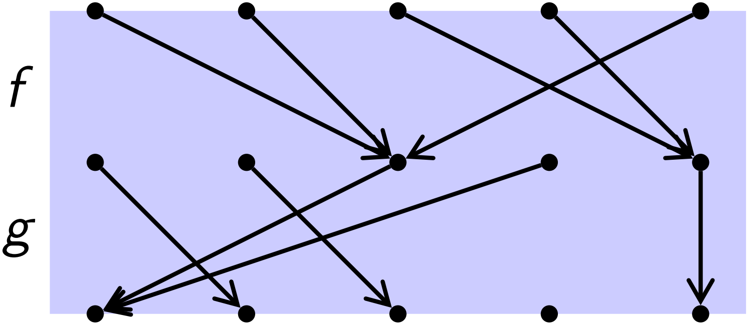
transformation

A *transformation* is a function from a set to itself, $f : X \to X$.



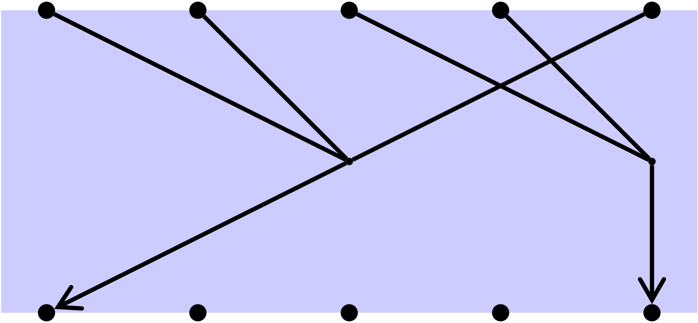
$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 3 & 5 & 5 & 3 \end{pmatrix}, \; X = \{1, 2, 3, 4, 5\}$$

We can combine transformations by stacking them...

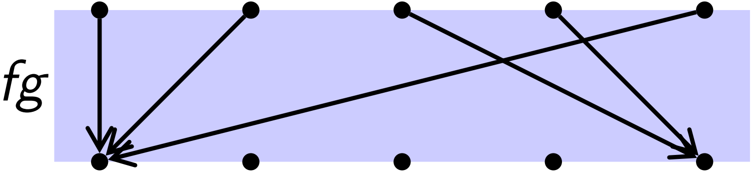


$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 3 & 5 & 5 & 3 \end{pmatrix}, g = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 1 & 5 \end{pmatrix}$$

...and then by following the connected lines...



...we get another transformation.



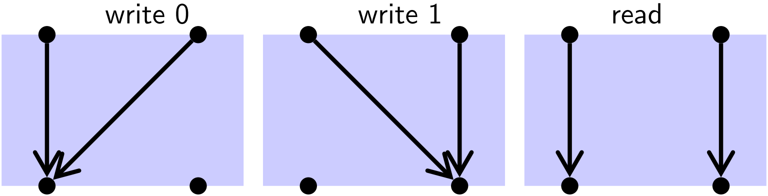
$$\begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 3 & 3 & 5 & 5 & 3\end{pmatrix} \cdot \begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 2 & 3 & 1 & 1 & 5\end{pmatrix} = \begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 1 & 1 & 5 & 5 & 1\end{pmatrix} = fg$$

Acting on the state set: $X^f = \{3,5\}$, $X^g = \{1,2,3,5\}$, $X^{fg} = \{1,5\}$

transformation semigroup

A *transformation semigroup* (X, S) of degree n is a collection S of transformations of an n-element set X closed under function composition.

Flip-flop, the 1-bit memory semigroup.



So these are computational devices... $\approx$ automata

btw.

	#subsemigroups	#conjugacy classes	#isomorphism classes
$\mathcal{T}_0$	1	1 (0)	1
$\mathcal{T}_1$	2	2 (1)	2
$\mathcal{T}_2$	10	8 (2)	7
$\mathcal{T}_3$	1 299	283 (4)	267
$\mathcal{T}_4$	3 161 965 550	132 069 776 (22)	TBA

A215650, A215651 `http://oeis.org`

`http://arxiv.org/abs/1403.0274`

hierarchical decompositions of finite transformation semigroups

	Natural Numbers	**Finite Automata**
Building Blocks	Primes	Flip-flop Automaton Permutation Automata
Composition	Multiplication	Cascade Product
Precision	Equality	Division, Emulation
Uniqueness	Unique	Different Decompositions

For computer algebra implementations we use the holonomy decomposition.

Approximating states – subset chains

Definition (Chain)

A *chain* c is a set of elements of $\mathcal{P}(X)$ with the property that for any $A, B \in c$ either $A \subseteq B$ or $B \subseteq A$.

Example (subset chains in general)

$X = \{1, 2, 3, 4, 5\}$

	$\{1,2,3,4,5\}$	
$\{1,2,3,4,5\}$	$\{1,3,4,5\}$	$\{2,3,4,5\}$
$\{2,4,5\}$	$\{1,3,4\}$	$\{3,4\}$
$\{4\}$	$\{3,4\}$	
	$\{4\}$	

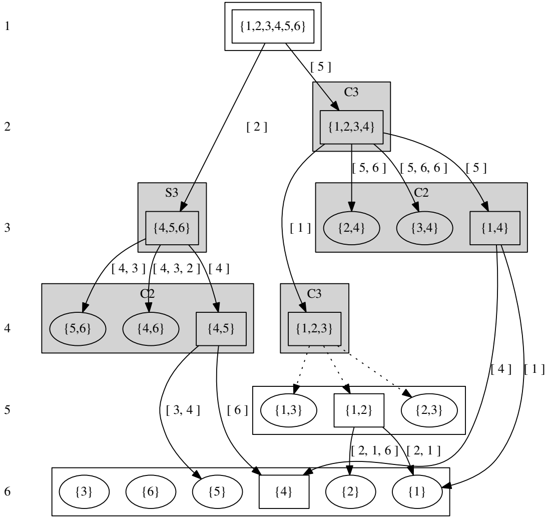
In the holonomy decomposition we act on subset chains, which needs a bit of orchestration.

extended image set

$\mathcal{I}(X) = \{X \cdot s \mid s \in S\}$

$\mathcal{I}'(X) = \mathcal{I}(X) \cup \big\{\{x\} \mid x \in X\big\} \cup X$

maximal chains in $\mathcal{I}'(X)$ are tile chains



tiles → reptiles → integers

abstract idea	**holonomy decomposition**
approximation	subset chains, tile chains, acting on tiles
compression	?
hierarchical structure	?

Compression – equivalence classes

Subduction relation

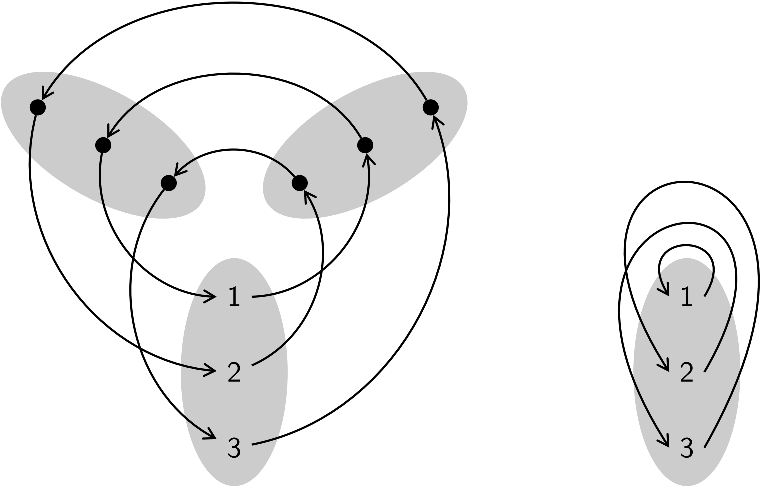
$$P \subseteq_M Q \iff \exists s \in M \text{ such that } P \subseteq Q \cdot s \quad P, Q \in \mathcal{I}(X), \tag{1}$$

i.e. we can transform Q to include P under the action of M. Therefore, subduction is a generalized inclusion, i.e. inclusion under the action of the trivial monoid.

$$P \equiv_M Q \iff P \subseteq_M Q \text{ and } Q \subseteq_M P.$$

$P \equiv_M Q \implies |P| = |Q|$ and $P = Qs$, $Q = Pt$ for some $s, t \in M$.

holonomy groups



(2,3)

holo what?

Definition (holonomy in differential geometry)

Given a smooth closed curve C on a surface M, and picking any point P on that curve, the holonomy of C in M is the angle by which some vector turns as it is parallel transported along the curve C from point P all the way around and back to point P.

(wiktionary.org)

abstract idea	holonomy decomposition
approximation	subset chains, tile chains, acting on tiles
compression	equivalence under action, encoding-decoding
hierarchical structure	?

hierarchical structure – cascade product

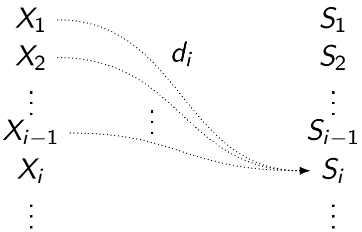
$(X_1, S_1), \ldots, (X_n, S_n)$
A state is an n-tuple $x = (x_1, \ldots, x_n)$, $x_i \in X_i$.

Definition

A *dependency function* d_i of level i for a list of components L is a function

$$d_i : X_1 \times \cdots \times X_{i-1} \to S_i, \quad i \in \mathbf{n}.$$

A dependency function of level i takes $i - 1$ arguments.



Definition

A *transformation cascade* for a given list of components is an n-tuple of dependency functions $(d_1, \ldots, d_n)$, where d_i is a dependency function of level i.

The action of a permutation cascade $d = (d_1, \ldots, d_n)$ on coordinates $x = (x_1, \ldots, x_n)$ for level i is defined by

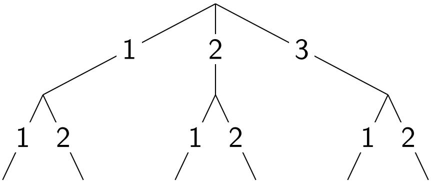
$$x_i^d := x_i^{d_i(x_1, \ldots, x_{i-1})}$$

thus the full action is

$$x^d = (x_1, \ldots, x_n)^{(d_1, \ldots, d_n)} = \left(x_1^{d_1(\varnothing)}, x_2^{d_2(x_1)}, \ldots, x_n^{d_n(x_1, \ldots, x_{n-1})} \right).$$

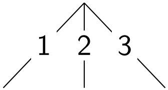
Coordinate tuples as paths in a tree

The set of coordinate tuples $\{(1,1),(1,2),(2,1),(2,2),(3,1),(3,2)\}$ can be represented by a tree.

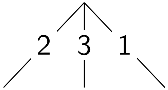


Acting on a tree

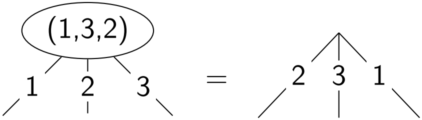
A permutation acting on points can be also thought as acting on a 1-level tree. For instance $(1, 3, 2)$ acting on



yields

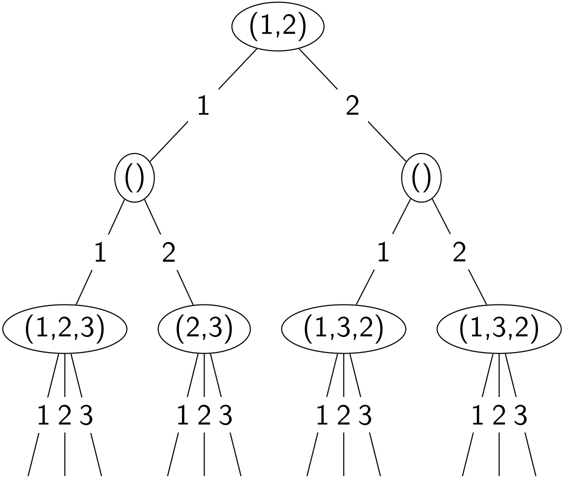


Now we can simply indicate this action on the branches by simply putting the permutation in the root node.

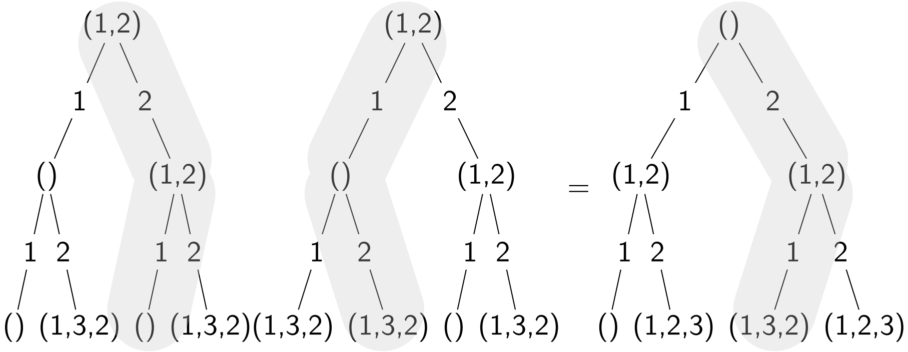


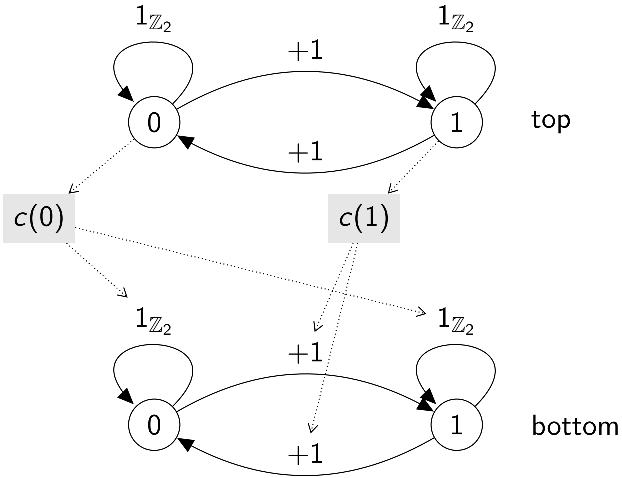
Permutation Cascade

Dependencies: $\varnothing \mapsto (1,2)$, $(1) \mapsto ()$ $(2) \mapsto ()$ $(1,1) \mapsto (1,2,3)$, $(1,2) \mapsto (2,3)$, $(2,1) \mapsto (1,3,2)$, $(2,2) \mapsto (1,3,2)$.



Multiplying Cascades




$1_{\mathbb{Z}_2}$
$1_{\mathbb{Z}_2}$
$+1$
0
1
top
$+1$
$c(0)$
$c(1)$
$1_{\mathbb{Z}_2}$
$1_{\mathbb{Z}_2}$
$+1$
0
1
bottom
$+1$

abstract idea	**holonomy decomposition**
approximation	tiling of subsets, tile chains
compression	equivalence under action, encoding-decoding
hierarchical structure	the placement of components, cascade product

ultimate goals

- what is computable with n states?
- how is the computation done?

Thank You!

Group & semigroup decomposition software:

SgpDec `http://sgpdec.sf.net`

On computational semigroup theory:

`http://compsemi.wordpress.com`