Ideas of the Holonomy Decomposition of Finite Transformation Semigroups

Attila Egri-Nagy[1] *and Chrystopher L. Nehaniv*[2]

June 10, 2017

[1]Akita International University, Japan,
`www.egri-nagy.hu`
[2] University of Hertfordshire, UK,
`homepages.herts.ac.uk/~nehaniv/`

A short, informal description of the intuitive ideas in the holonomy decomposition of finite transformation semigroups.

Calculus versus automata theory

One of the fundamental concepts of science and computation is the notion of *change*: a system goes from a state to another state due to external manipulations or due to internal processes at various time-scales. If the set of states is a continuum then we study continuous functions and thus we do analysis. If we have a set of discrete states then we do automata theory from theoretical computer science. Since automata theory has an algebraic level of description, we end up in abstract algebra, namely in semigroup theory.

Transformation semigroup

A *transformation semigroup* (X, S) captures the concept of change in a rigorous and discrete way. It consists of a set of *states* X (analogous to *phase space*), and a set S of transformations of the state set, $s : X \to X$ acting by $x \mapsto x \cdot s$, that is closed under the associative operation of function composition. Writing $s_1 s_2 \in S$ for the composite function $s_1 \in S$ followed by $s_2 \in S$, we have $x \cdot (s_1 s_2) = (x \cdot s_1) \cdot s_2$, giving a (right) *action* of S on X. Transformation semigroups are general enough to model a wide range of processes. All we need is to have a strong structure theorem for them.

Finite state automata (without specifying initial and accepting states) and transformation semigroups are essentially the same concepts, since a fixed generating set for a transformation semigroup can be considered as a set of input symbols.

Decompositions

Another fundamental technique of the scientific method is *decomposition*. We identify the building blocks of a system, and how these components work together to build the system. The simpler components are easier to understand. We gain more understanding from the decomposition if these connections are somehow limited. If the information goes only in one direction, we talk about a *hierarchical* system. The least dependent component does not receive any information from others, while components deeper in the hierarchy are influenced by the building blocks above.

Krohn-Rhodes theory

It is a remarkable result of finite semigroup theory [7], that we can always find a decomposition in a hierarchical form. There is a caveat though, we often end up building a bigger system through hierarchical composition. So instead of two systems being the same, we need to talk about *emulation*, which is in general a capability of one system producing the same dynamics as another one, not necessarily containing an exact copy. For semigroups, we say that S *divides* T, if S is a homomorphic image of a subsemigroup of T.

Algebraically, hierarchical connections are captured by wreath products. Now we can state a main result of algebraic automata theory.

The *wreath product* $(X,S) \wr (Y,T)$ of transformation semigroups is the transformation semigroup $(X \times Y, W)$ where

$$W = \{(s,f) \mid s \in S, f \in T^X\},$$

whose elements map $X \times Y$ to itself as follows

$$(x,y) \cdot (s,f) = (x \cdot s, y \cdot f(x))$$

for $x \in X, y \in Y$. Here T^X is the semigroup of all functions f from X to T (under pointwise multiplication). Note we have written $y \cdot f(x)$ for the element $f(x) \in T$ applied to $y \in Y$. The wreath product construction is associative on the class of transformation semigroups (up to isomorphism) and can be iterated for any number of components.

Theorem 1 (Krohn-Rhodes Theorem (informal))**.** *Every finite semigroup S is a divisor of wreath product of its building block components. The groups in the components are divisors of S itself.*

This is analogous to the Jordan-Hölder Theorem in group theory, but there we can use embedding instead of division.

The holonomy method

The holonomy decomposition is one particular method for finding the building blocks of transformation semigroups and composing them in a hierarchical structure. Beyond the ideas of emulation and hierarchy, we need two more fundamental concepts: *approximation* and *compression*.

Approximation gives less information about a system in a way that the partial description does not contradict the full description. In the holonomy decomposition, we extend the action to be defined on sets of states. Thus, a state is approximated by a set containing it. Then, we further extend the action to chains of increasingly smaller subsets of the state set, that successively approximate a state. The hierarchical nature of the decomposition also originates in these nested sets. The technical details of the holonomy method are for putting the extended action on chains into the form of a wreath product.

To do this we need compression, that for repeated patterns stores the pattern once and then only records its occurrences. Whenever the semigroup acts the same way on different subsets, we consider those subsets equivalent and only store the action on the equivalence class representatives (compression). These representative local actions are the building blocks of the decomposition, and they are permutation groups augmented with constant maps. They can be defined by round-trips of mappings of elements of the equivalence

The easiest example is the holonomy decomposition of the semigroup of all maps of an n-element set, the *full transformation semigroup* $\mathcal{T}_n$. It divides a cascade product with $n-1$ levels, on each level with a symmetric group $\mathcal{S}_m$ with constants, $2 \leq m \leq n$.

classes. The term 'holonomy' is borrowed from differential geometry: a round-trip of composed bijective maps producing permutations is analogous to moving a vector via parallel transport along a smooth closed curve yielding change of the direction of the vector.

For the complete algorithm see [3].

Outlook

There are many possible applications of Krohn-Rhodes theory [6]. Most recently, it has been picked up in the study of complex systems, under the general concept of renormalization, coming form theoretical physics and information theory [1]. However, before the theory can realize its full potential, we still need to further mathematical research about its computational aspects. For computational experiments, a software package called SgpDec [2, 4] is available for the GAP computer algebra system [5].

References

[1] Simon Dedeo. Introduction to Renormalization – online course. `renorm.complexityexplorer.org`. Santa Fe Institute, 2017.

[2] Attila Egri-Nagy, James D. Mitchell, and Chrystopher L. Nehaniv. Sgpdec: Cascade (de)compositions of finite transformation semigroups and permutation groups. In *Mathematical Software ICMS 2014*, volume 8592 of *LNCS*, pages 75–82. Springer, 2014.

[3] Attila Egri-Nagy and Chrystopher L. Nehaniv. Computational Holonomy Decomposition of Transformation Semigroups. arXiv:1508.06345 [math.GR], 2013.

[4] Attila Egri-Nagy, Chrystopher L. Nehaniv, and James D. Mitchell. SgpDec – *software package for Hierarchical Composition and Decomposition of Permutation Groups and Transformation Semigroups, Version 0.8+*, 2017. `gap-packages.github.io/sgpdec/`.

[5] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.7.8*, 2017.

[6] John Rhodes. *Applications of Automata Theory and Algebra via the Mathematical Theory of Complexity to Biology, Physics, Psychology, Philosophy, and Games*. World Scientific Press, 2009. Foreword by Morris W. Hirsch, edited by Chrystopher L. Nehaniv (Original version: UC Berkeley, Mathematics Library, 1971).

[7] John Rhodes and Benjamin Steinberg. *The q-theory of Finite Semigroups*. Springer, 2008.