Categories, Mathematics, and Systems

Larry A. Lambe[1], PhD, PhD (Hon)
MSSRC[2]
llambe@mssrc.com
llambe@acm.org
http://www.mssrc.com

June, 2018

[1]Chief Scientist, Multidisciplinary Software Systems Research Corporation

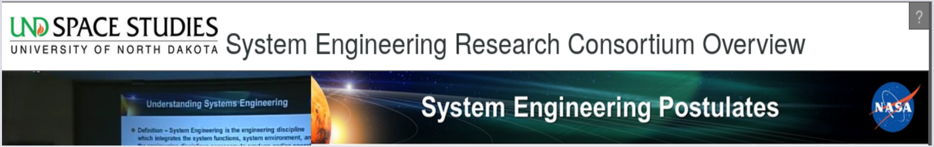
[2]The speaker is grateful to Professor Ronnie Brown, Bangor University, Wales, UK for many years of useful conversations about mathematics

1 **TBA**

Introduction

- A Webinar with the same title as this one was given in January of this year for system engineers (INCOSE and ISSS) in March.
- Details and a video of th Webinar are available at <http://mssrc.com/lambe/notes_se_cats.html>.
- The goal was to provide a background for understanding language that is used in a growing number of works dealing with ideas about systems in terms of what is called **category theory**.
- A knowledge of category theory is not assumed here. Basic ideas from that subject will be presented in an intuitive way.
- We will begin with a couple of Postulates given by Dr. Michael D. Watson of the NASA System Engineering Office NASA Marshall Space Flight Center.

Michael D Watson: http://space.edu/colloquium.aspx



- **Postulate 2: The Systems Engineering domain consists of subsystems, their interactions among themselves, and their interactions with the system environment**
- **Postulate 7: Understanding of the system evolves as the system development or operation progresses**

Objectives

- Mention some preliminary mathematical notions (set theory, directed graphs, algebra)
- Give some examples
- Present category theory basics with more examples
- Give a **working definition** of "system" in terms of the language of category theory using intuition and Postulate 2
- Very briefly mention Postulate seven in light of the previous discussion

Mathematical Categories

- In this talk, we will deal only with what are called "small categories".
- Small categories are collections whose parts are contained in a **set** of things in the sense of ordinary set theory.
- From now on when we use the word "category" we will mean small category.
- A category can be defined in terms of a special kind of directed graph (digraph).
- A directed graph in turn may be defined in terms of nodes and directed edges (*arrows*).
- The directed graph associated to a category must have a special kind of arrow associated with each node. That special arrow is called the *identity* arrow for that node.

A One Node Two Loop Category

- A simple digraph that has only one node and two arrows is presented as

Figure: One node; one arrow, label a

- Only one arrow is displayed (the identity is not labeled). We may define the operation of "going around the loop" (a *path*) one time and represent that by the label "a".

A One Node Two Loop Category

- We denote going around twice by “aa” an so on.
- Thus going around four times would be denoted by ‘aaaa”.
- The operation of staying put at the node will acquire the label of the identity arrow. Clearly, the set of all possible paths in the graph above is given by

$$M = \{id_\bullet, a, aa, aaa, \ldots, a^n, \ldots\} \quad (1)$$

- From a mathematical viewpoint, this set with its path product and identity path forms a **monoid**
- i.e. a set with operation that is associative and has an identity element.

A One Node Three Loop Category

- This time, we have one node and two arrows labeled “0” and “1”, and thus a function defines a decomposition of its source X.
- Again, the identity arrow is labeled $id_\bullet$. but not displayed.

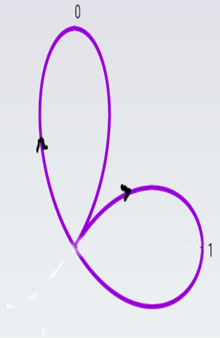


Figure: One node and three arrows (compose by juxtaposition).

A One Node Three Loop Category (Cont'd)

- This time, the set of all path products are all "words" in 0 and 1.
- So we have compositions like $0100100001100101011011000110110001101111$ and $0111011101100111011100100110110001100100$
- which incidentally, spell 'Hello world' in English using the ASCII encoding.
- In fact, all of recorded history is encoded in this little category!

More Categories

- We can give a definition at this point which should make sense in terms of what has been presented so far.
- *A category* $\underline{C}$ *is a directed graph whose nodes are one loop monoids.*
- Note that it is assumed that all composite paths are considered as being in the category.
- A further condition is that **relations** may be allowed among the arrows.
- For example, even in the simply one node two arrow category, we may impose the relation $a^2 = id_\bullet$.
- Generally, the relations in a category can be quite complicated and there are mathematical theories that deal with this situation, but they are beyond the scope of this presentation.
- See <http://mssrc.com/lambe/notes_se_cats.html> for more information.

An I/O Chain Category

- In aerospace engineering, a familiar situation is one involving what is called a *black box* procedure.
- A situation which arises in the normal course of multidisciplinary analysis (MDA) involves a chain of black box situations.
- If we denote a link in such a chain by the following diagram

$$\mathcal{BB} = [\mathcal{I} \rightarrow \boxed{\textbf{Black Box}} \rightarrow \mathcal{O}]$$

- where $\mathcal{I}$ is input and $\mathcal{O}$ is output.
- The problem of 'I/O chains' is the following.
- Given a sequence of black boxes $\mathcal{BB}_1, \ldots \mathcal{BB}_m$, find a way to compose the chain
- $\mathcal{BB}_1 \rightarrow \mathcal{BB}_2 \rightarrow \ldots \mathcal{BB}_m$.

An I/O Chain Category (Cont'd)

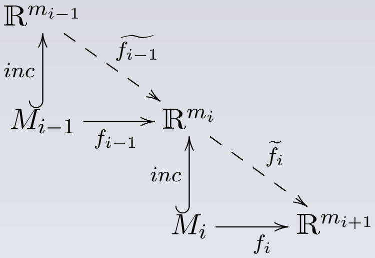
- Assumption: All inputs and outputs are subsets of some universal set $U = \mathbb{R}^n$ for n sufficiently large.
- For example, the first black box may be a solver in the area of aerodynamics,
- the second, in propulsion systems, the third in thermodynamics, etc.
  - **A Current Strategy:**
  - Use some method, e.g. interpolation to extend the input domain $\mathcal{I}_{i+1}$ so that outputs from $\mathcal{O}_i$ can be input to $\mathcal{BB}_{i+1}$.
- A good solution is a statistical model and a precise probabilistic measure of goodness such as a globally defined mean squared error.
- A simple digraph describes the situation, viz.

An I/O Chain Solution

$$\bullet_1 \xrightarrow{f_1} \bullet_2 \xrightarrow{f_2} \bullet_3 \xrightarrow{f_3} \bullet_4 \xrightarrow{f_4} \bullet_5 \xrightarrow{f_5} \bullet_6 \xrightarrow{f_6} \cdots$$

- The nodes represent triangulation of spaces upon which the indicated solvers are defined.
- However, the arrows are *not* functions between the nodes because the solver at node i may have output that does not fall on the mesh defined at node $i+1$.
- So what are the arrows?
- We use a diagram to illustrate that arrows need not be functions and, in fact, can be quite involved in nature.

An I/O Chain Solution (Cont'd)



- The node $\bullet_i$ is the mesh M_i and the arrow is the triangular system you see above for M_i.
- The f_i are thought of as "training sets" and the $\widetilde{f_i}$ are Gaußian process regression models based on the f_i
- This is a typical MDA situation, but there are many other scenarios.

Back to One Node Examples: Functors

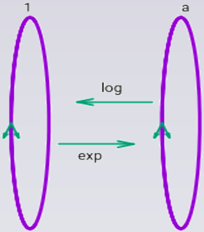


Figure: Left, node = 0, arrow = 1, right, node = 1, arrow = a

- By what we have presented so far, one can see that the monoid at the left has path products $0, 1, 2, \ldots, n \ldots$ and an operation which we can denote by $+$. The one on the right has path products $1, a, a^2, \ldots, a^n, \ldots$ and an operation of juxtaposition which satisfies $a^m a^n = a^{m+n}$.

Back to One Node Examples: Functors (Cont'd)

- These monoids are, in fact isomorphic (they are one-to-one and preserve path products). The isomorphisms are given ion nodes by

$$\log_{\mathrm{a}} : 1 \mapsto 0$$

$$\exp_{\mathrm{a}} : 0 \mapsto 1$$

- **and on arrows** by

$$\log_{\mathrm{a}}(a^n) = n$$

$$\exp_{\mathrm{a}}(n) = a^n.$$

- Note that functors must be correspondences between nodes **and** arrows.
- They must also preserve path product which these two well-knows correspondences do!

An Intuition

- *A colimit is intuitively an object that has been obtained by "machining together" a collection of parts using a collection of "compatibility/consistency" instructions.*
- Before we can formalize this notion, we need a more precise definition of "a collection of parts".
- For that, we define *a diagram in a category $\underline{D}$ indexed by a category $\underline{C}$ and over a functor $F : \underline{C} \to \underline{D}$ is the image of such a functor*.
- **Definition:** A cocone in $\underline{D}$ is a correspondence between a diagram in $\underline{D}$ and an object (node) in D.
- There should be arrows from each node in the diagram to the object D.
- All sub-diagrams should "commute",

An Example Cocone

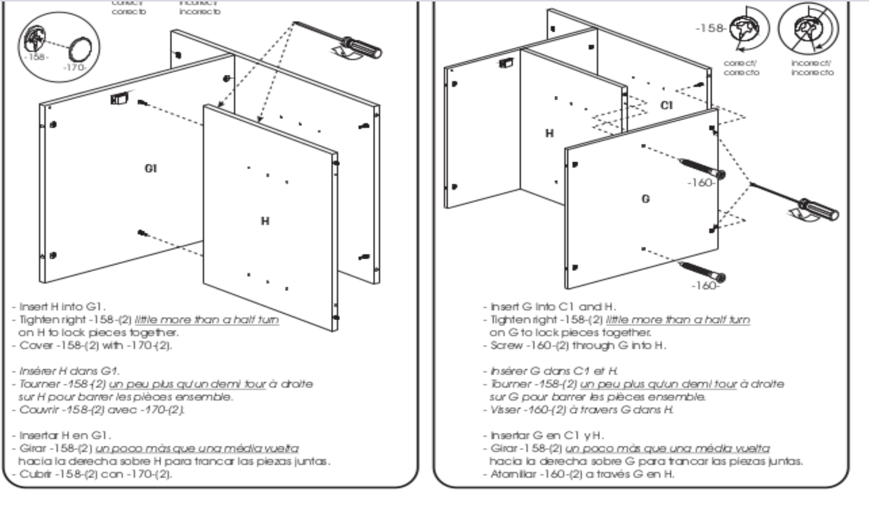


Figure: From some printer stand instructions

A Working Definition of System

- To allow for finite recursion in our definition, we mark a cocone *atomic* if all the nodes comprising it except the vertex are considered to be indecomposable, i.e. not vertexes of cocones themselves.

Definition

A system is the vertex of either an atomic final cocone or a final cocone whose nodes are recursively systems.

- This definition was inspired by Watson's postulates 2 and 7. Its relation to postulate two should be clear, postulate seven could be addressed by allowing nodes to be petri nets which are known to be certain kinds of categories.
- These notions are under discussion in a Consortium consisting of NASA and a number of Universities and Labs.