

# Modeling Open Systems with Category Theory

---

Daniel Sinderson

2024

Southern Oregon University

# Objective and Scope of the Project

---

## Goal of the Project

My objective for this project was to learn about the field of categorical systems theory, which applies the pure math field of category theory to the study of arbitrary systems, and then use what I learned to model a real-world system.

# The Scope of the Project

1. Learn category theory
2. Learn categorical systems theory
3. Learn about a real world system  
(transcription networks)
4. Learn how to model and numerically solve it  
(Julia language)

# Scope of this Presentation

1. Tell you what a system is.
2. Tell you what a category is.
3. Tell you how they can work together.
4. Show you a system and how I modeled it.
5. Talk about the simulated results.

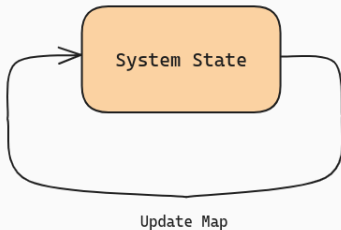
# Systems

---

# What is a System?

A system is a thing that changes. At its simplest, a system consists of the following two things.

1. A set of states that the system can be in.
2. A map that updates the system's state based on the state that it's currently in.



## Examples of Systems

The set of natural numbers with the successor function is a system.

$$S : \mathbb{N} \rightarrow \mathbb{N}$$

$$n \mapsto n + 1$$



An ordinary differential equation is also a system.

$$\frac{dx}{dt} = \kappa x$$

Here the set of states is the set of real numbers  $\mathbb{R}$  and the update map is the differential equation itself given some initial condition  $x_0$ .

## What's the Problem

The problem with these systems is that they're closed.

They don't interact with each other, and they don't interact with their environment.

So let's open them up with category theory.

# Categories

---

Category theory is a theory of associative composition.

It's very general, and this generality allows it to encapsulate much of the high level structure in other theories.

This gives us a common formalism for working with the abstract structural features found across a multitude of diverse phenomena, both inside and outside of pure mathematics.

# What is a Category?

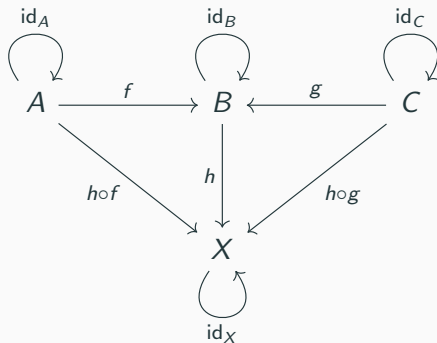
## Definition (Category)

A category  $\mathcal{C}$  is defined by the following:

1.  $\mathcal{C}$  contains a collection of objects  $\text{ob}(\mathcal{C})$ .
2. For any two objects  $a, b \in \mathcal{C}$  there is a collection of morphisms,  $f : a \rightarrow b$ , between those objects called the homset,  $\mathcal{C}(a, b)$ .
3. Every object  $a \in \mathcal{C}$  has a morphism to itself  $\text{id}_a : a \rightarrow a$  called its identity.
4. For every two morphisms  $f : a \rightarrow b$  and  $g : b \rightarrow c$  there's a third morphism  $g \circ f : a \rightarrow c$  that's their composition.

# What is a Category?

## Definition (Category)



# What is a Category?

These objects and morphisms are then under two constraints:  
unitality and associativity.

**Definition (Category cont. Unitality)**

Any morphism  $f : a \rightarrow b$  can be composed with the identity morphisms of  $a$  and  $b$  such that  $f \circ \text{id}_a = \text{id}_b \circ f = f$ .

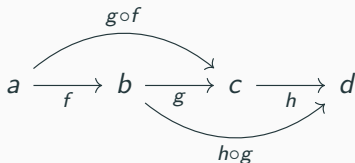
$$\text{id}_a \circ a \xrightarrow{f} b \circ \text{id}_b$$

# What is a Category?

These objects and morphisms are then under two constraints:  
unitality and associativity.

## Definition (Category cont. Associativity)

For any morphisms  $f : a \rightarrow b$ ,  $g : b \rightarrow c$ , and  $h : c \rightarrow d$ ,  
 $h \circ (g \circ f) = (h \circ g) \circ f$ . Since it doesn't matter what order we  
apply the morphisms, we write this  $h \circ g \circ f$ .



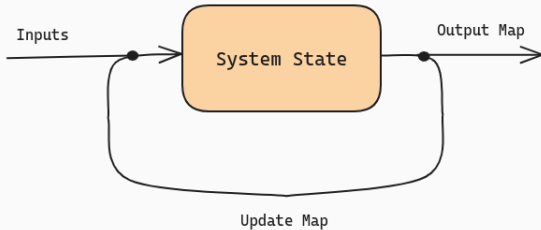


# Composing Systems

---

# What is a Open System?

An open system is as a system with an interface.



# Lenses and Arenas

To model this mathematically we'll use objects called arenas and morphisms called lenses.

## Definition (Arenas)

An arena is a pair of objects  $\begin{pmatrix} A \\ B \end{pmatrix}$  from an underlying category.

## Definition (Lenses)

A lens is a morphism between arenas made from a pair of morphisms from the underlying category

$$\begin{pmatrix} f \\ g \end{pmatrix} : \begin{pmatrix} A \\ B \end{pmatrix} \rightleftarrows \begin{pmatrix} C \\ D \end{pmatrix}$$

where  $g : B \rightarrow D$  is the output map and  $f : B \times C \rightarrow A$  is the update map.

## What this Gives Us

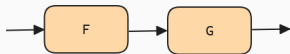
Arenas and lenses between them form a monoidal category.

This means that we can compose systems and multiply them.

By modeling our open systems as lenses, composition of lenses lets us place systems in series and multiplication of lenses lets us place systems in parallel.

# Systems in Series and Parallel

Systems in Series:  $\begin{pmatrix} g^\# \\ g \end{pmatrix} \circ \begin{pmatrix} f^\# \\ f \end{pmatrix}$



Systems in Parallel:  $\begin{pmatrix} g^\# \\ g \end{pmatrix} \otimes \begin{pmatrix} f^\# \\ f \end{pmatrix}$



## Case Study

---

# Gene Transcription

Gene transcription is the process by which genes in a cell's DNA produce proteins.

The dynamics of this process for a single gene are well modeled by the following differential equation.

$$\frac{dY}{dt} = \beta f(X^*) - \alpha Y$$

$$\text{Activator: } f_A(X^*) = \frac{\beta X^{*n}}{K^n + X^{*n}} \quad \text{or} \quad \beta * L(X^* > K)$$

$$\text{Repressor: } f_R(X^*) = \frac{\beta}{1 + \left(\frac{X^*}{K}\right)^n} \quad \text{or} \quad \beta * L(X^* < K)$$

# Motor Flagella Network

Gene transcription network are connected networks of genes whose products act as transcription factors for other genes.

Below is the motor flagella network of *e. coli* bacteria.

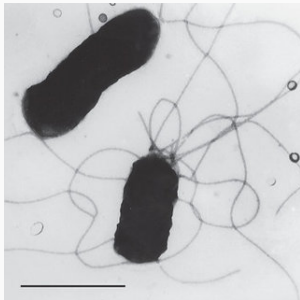
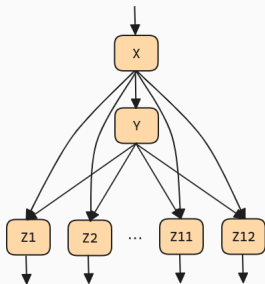
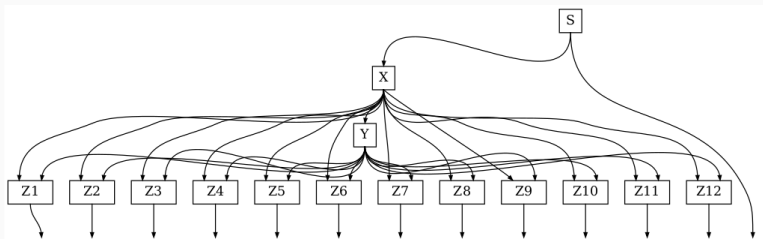


Image from *Bacterial Chemotaxis* by Michael Eisenbach



# Motor Flagella Network Model

Using lenses we can wire together the transcription factor dynamics as differential systems and then simulate the results.



# Motor Flagella Network Model

$$\text{Gene X: } \frac{dX}{dt} = [\beta * L(S > \kappa_S) - \alpha * X]$$

$$\text{Gene Y: } \frac{dY}{dt} = [\beta * L(X > \kappa_X) - \alpha * Y]$$

$$\text{Gene Z1: } \frac{dZ_1}{dt} = [\beta * L(X > \kappa_X \text{ OR } Y > \kappa_Y) - \alpha * Z_1]$$

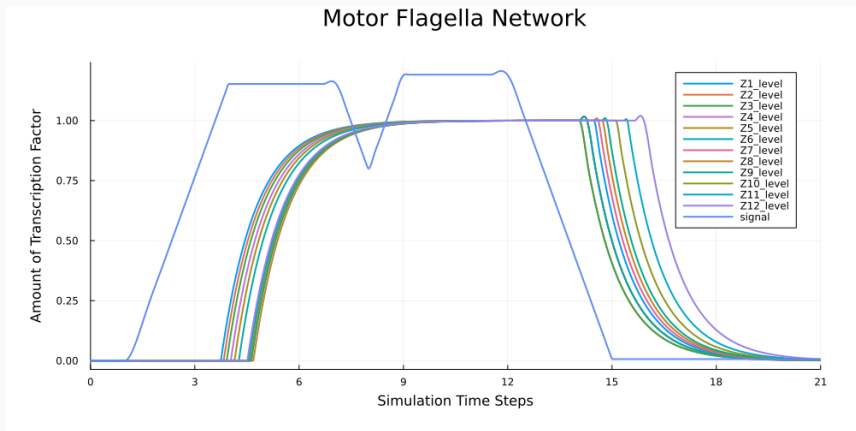
$$\text{Gene Z2: } \frac{dZ_2}{dt} = [\beta * L(X > \kappa_X \text{ OR } Y > \kappa_Y) - \alpha * Z_2]$$

$\vdots$

$$\text{Gene Z11: } \frac{dZ_{11}}{dt} = [\beta * L(X > \kappa_X \text{ OR } Y > \kappa_Y) - \alpha * Z_{11}]$$

$$\text{Gene Z12: } \frac{dZ_{12}}{dt} = [\beta * L(X > \kappa_X \text{ OR } Y > \kappa_Y) - \alpha * Z_{12}]$$

# Motor Flagella Network Simulation Results



Simulation results were obtained using the numerical solver in the OrdinaryDiffEq.jl library for the Julia programming language.

## Conclusion

---

## Where did all the Category Theory Go?

It's inside the wiring diagrams.

Diagrammatic languages are more intellectually tractable.

They are easier to reason about for both the designer or modeler and those they are designing or modeling for.

# What's Next?

Lenses over other cartesian categories: **Set**, **Man**

Adding monads for nondeterministic systems: MDP, SDE

Using charts and double categories to compose system behavior as well as specification

Thank You!



ALON, U.

**An introduction to systems biology: design principles of biological circuits.**

Chapman and Hall/CRC, 2019.



FONG, B., AND SPIVAK, D. I.

**An invitation to applied category theory: seven sketches in compositionality.**

Cambridge University Press, 2019.



MYERS, D. J.

**Categorical systems theory.**

*Manuscript in preparation (2022).*