Week Seven

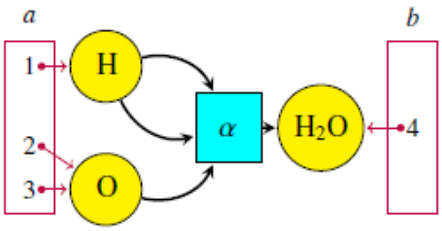
Siva Sundar, EE23B151

August 2024

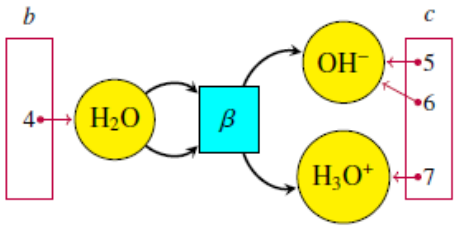
25th August

Open Petri-net

- These contain **open** places which interact with other petri-nets outside it.

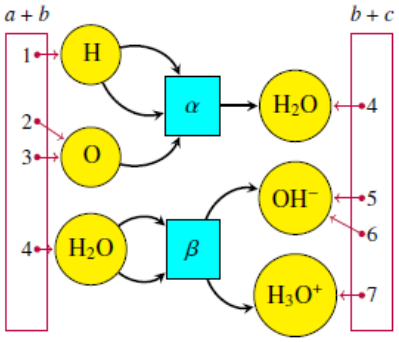


Petri net representing $2H+O \longrightarrow H_2O$



Petri net representing self-ionization.

- We use extra objects whose elements get mapped to these open places. This creates a sense of '**port**' and ports of the same type (element) can be connected together.

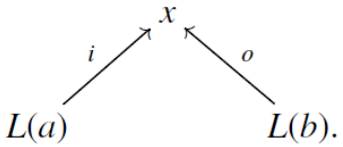


Combined Petri net

- When plugged together, the contents of the open place are shared between the two Petri nets.
- An open petri net is a *structured cospan*.

Structured Cospan

It is a mathematical object which is of the form:



- L: A *functor* mapping the object A to the combined Petri net X.
- A: The object representing the set of pre-images for both input and output places.

- X: The combined Petri net that results from the structured composition.
- i and o are called the **legs** of the cospan. They map the input and output places to the closed petri-net x.
- a and b: Subsets of A representing the pre-images of the input and output places, respectively. These subsets are referred to as the **input** and **output** sets.
- x: The specific part of the Petri net, also known as the **apex**, which is determined by the functor L. The functor L maps a to the input places and b to the output places of this apex.

Given the input set a and output set b, the apex x is the portion of the Petri net where a and b correspond to the input and output places, respectively.

In the context of chemical reactions, such as the water formation reaction, the sets a and b correspond to the **pre-images** of the input and output places, respectively. Specifically:

- The set a represents the **inputs** to the water formation reaction.
- The set b represents the **outputs** of the water formation reaction.

The *functor* L maps these sets to the corresponding elements in the Petri net:

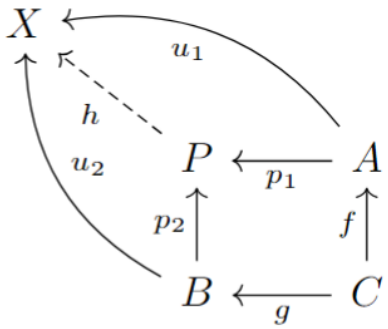
- $L(a)$ maps the elements in a to the open input places in the Petri net.
- $L(b)$ maps the elements in b to the open output places in the Petri net.

Using the mappings $L(a)$ and $L(b)$, we can extract the Petri net x that models the water formation reaction. This Petri net x incorporates the connections and interactions defined by $L(a)$ and $L(b)$, thereby representing the complete chemical reaction process.

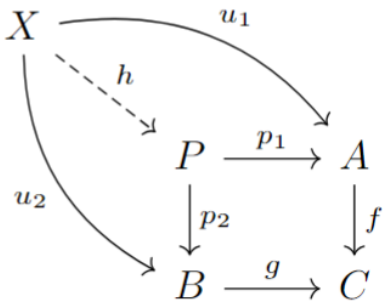
26th August

Pushouts and Pullbacks

Pushouts and Pullbacks are objects which have the universal property that for all objects X, both the inner and outer squares commute in each figure:



Here, P is **Pushout**



Here, P is **Pullback**

Speaking in terms of sets, we can relate Pushouts to the Union of the individual sets and Pullbacks as Intersection of those sets.

Started learning from Navin's petri net code and tried making my own model (not finished yet). **CLICK HERE** to see *jupyter notebook* which contain the code.

1st September

Read the remaining part of paper which includes advantages of using petri-nets, typed petri-nets and stratification of models (skipped the caliberation section).

Advantages of using Petri nets for modelling:

- In operadic modeling, the structure of a model is made clear and directly translates from math to code, unlike traditional software where this structure is hidden in the code.
- A submodel may itself be the composition of still more primitive models, and thus, making this a hierarchial model.
- Can reduce code complexity and errors, since local changes to models correspond to local changes in the code base.
- As this approach makes a model's theory clearer and closely linked to its code, modelers can spot unnecessary complexities, inaccurate domain representations (model doesn't correctly reflect the real-world system), or overly strong assumptions.
- Since the rules for combining model components are clearly defined and separate, experts can easily share knowledge through the composition syntax, while the selection of specific submodels can be handled by domain experts or a model selection process.
- The composition syntax can be analyzed for mechanistic or causal dependencies.
  eg: In the Ross-Macdonald model, host and vector can only interact via the bloodmeal.
- The associativity and symmetry properties of operads and their algebras imply that the order of composing submodels does not affect the final model.

Typed Petri-Nets

- Morphisms between Petri nets can be used to define typed Petri nets. A morphism between Petri nets is a map of places, transitions, input arcs, and output arcs that preserves the arities of the arcs (indicates how many places are connected to a particular transition.) and respects the sources and targets of the arcs. These maps are also called **etale maps**.
- A **type system** is a formal set of rules in programming and computer science that assigns properties or types to various constructs in a computer program, such as variables, expressions, functions, or modules.
- Petri nets can define domain-specific type systems. For example, the Petri net in Figure 4(a) (of research paper) defines a type system for an infectious disease model with one species type and three transition types: (1) changes in infection status, (2) non-infection-related changes, and (3) interactions between individuals.
- All of the Petri nets typed by a given type system form a slice category (see ref. [1]) of the category of whole-grain Petri nets.

The "Category of **whole-grain** Petri nets" likely refers to a category where:

- ⋆ Objects are Petri nets.
- ⋆ Morphisms are **structure-preserving** maps between Petri nets.

In nutrition, "Whole-grain" refers to grains that **retain** all of their original components. Here, "Whole-grain" suggests these Petri nets **retain** all their structural information.

Stratification

Stratification refers to a method of creating more complex models by replicating local dynamics across multiple "strata" or layers, and then specifying how these strata interact by a 'scheme'.

To stratify a model $P \longrightarrow P_{type}$, we need a 'stratification scheme' (see fig (5) in research paper). This scheme is a category $P_{strata} \longrightarrow P_{type}$. Here,

- P represents the original petri net model which we want to stratify.
- P_{strata} represents the 'scheme' or method by which we need to 'stratify' or layer the individual places in P.
- P_{type} represents the type systems common to P and P_{strata}. ie, It helps to identify the places in P which are valid for stratification by P_{strata}.

The above points can be visualised using the figure on the right. We can see that $P_{stratified}$ (the stratified petri-net of model 'P' by 'P_{strata}') is indeed, a **pullback**. Also, the size of stratified models increases **quadratically** because each place, transition, and arc in the stratified model is a combination of elements from the component models.

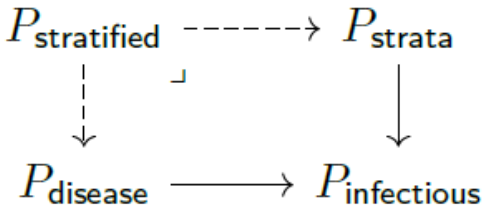


Figure 1: Stratification as a pullback of model 'disease' by 'strata' which share type of 'infectious'. (see fig.6 in paper)

While using multiple stratas for stratification, the order in which stratification is done doesn't matter as pullbacks are commutative and associative.

References

[1] ProofWiki: Slice Category