Compositional design and tasking of networks

John D. Foley
Metron, Inc.
foley@metsci.com

—

(joint w/ J. C. Baez, J. Moeller, and B. S. Pollard)

2017 AMS Sectional Meeting at UCR

Depending on the application, various kinds of networks provide useful models:

2-way communication	simple graphs
1-way communication	directed graphs
multiple channels	multigraphs
computer programs	Petri nets

and we often want to build up complex networks from simple ones.

Instead of separately studying each kind of network, we introduce:

- the concept of 'network model' and
- a functor to convert each network model to a **typed operad** whose purpose is to compose many smaller networks into a single network.

Constructing network operads

- Start with your favorite lax symmetric monoidal functor $F : \mathbf{S} \to \mathbf{Cat}$
- apply the symmetric monoidal Grothendieck construction to get the symmetric monoidal category $\int F$ with $\otimes_F$
- Let O_F be the endomorphism operad $\mathrm{op}(\int F)$

Theorem

The composite

$$\mathbf{NetMod} \xrightarrow{\int} \mathbf{SSMC} \xrightarrow{\mathrm{op}(-)} \mathbf{Op}$$

constructs a network operad for each network model.

Network operads for this talk

- In the next talk, Joe Moeller will talk about the construction of network operads O_F from network models F.
- In this talk, we give examples of network operads O_F that act on algebras A to compose k networks into one.
- All O_F will come from $F\colon \mathbf{S} \to \mathbf{Mon}$, where $\mathbf{S} = \coprod S_n$ is the groupoid of all finite permutations and **Mon** is the category of monoids.
- This will allow us quickly describe O_F and go from there.
- After showing the definition operations in O_F, we will suppress the role of symmetries in our examples.

Typed operads and their algebras

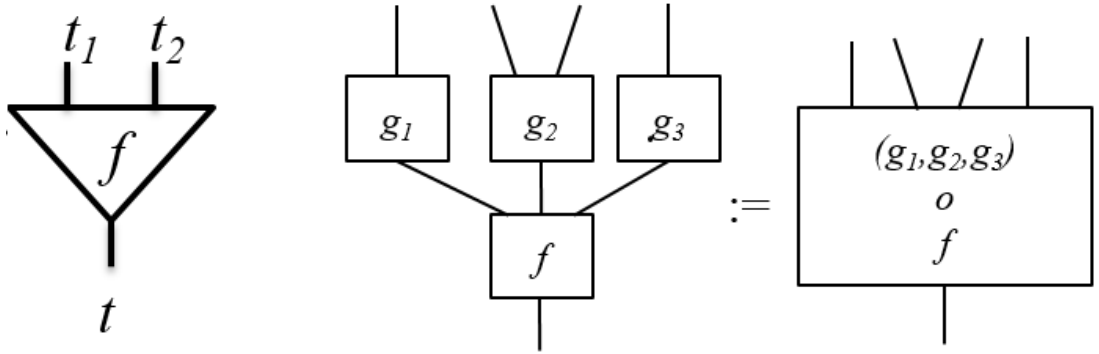
Recall that a typed operad O has:

- a set T of **types**,
- sets of operations $O(t_1, ..., t_n; t)$ where $t_i, t \in T$,
- ways to composes operations

$$f \circ (g_1, \ldots, g_n) \in O(t_{1i}, \ldots, t_{1k_1}, \ldots, t_{n1}, \ldots t_{nk_n}; t),$$

- and ways to permute the arguments of operations,

which obey some rules [6].



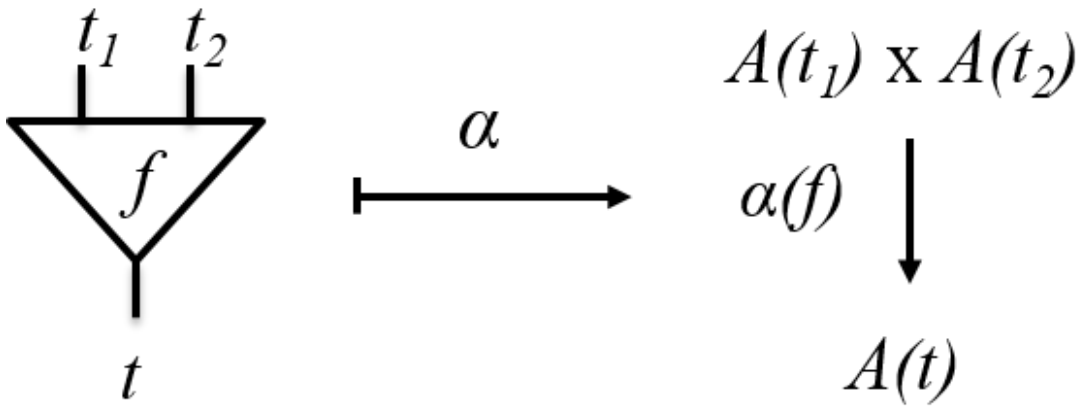
Typed operads and their algebras

An algebra A of O specifies a set of things of each type $t \in T$ such that the operations of O act on these sets. Thus, it has:

- for each type $t \in T$, a set $A(t)$ of things of type t,
- ways to apply any operation f to things a_i to obtain a thing

$$\alpha(f)(a_1, \ldots, a_n) \in A(t).$$

Again, we demand that some rules hold [6].



Operads and algebras for this talk

For our network operads O_F, coming from $F\colon \mathbf{S} \to \mathbf{Mon}$, the sets of operations are:

$$\mathrm{O}_F(n_1, \ldots, n_k; n) = \begin{cases} S_n \times F(n) & \text{if } n_1 + \cdots + n_k = n \\ \emptyset & \text{otherwise.} \end{cases}$$

The monoids $F(n)$ will parameterize networks with n vertices and the monoid operation in $F(n)$ describes how to 'overlay' one network with n vertices on top of another.

For each algebra A of O_F, the sets $A(n)$ will be the networks with n vertices we wish to compose with O_F.

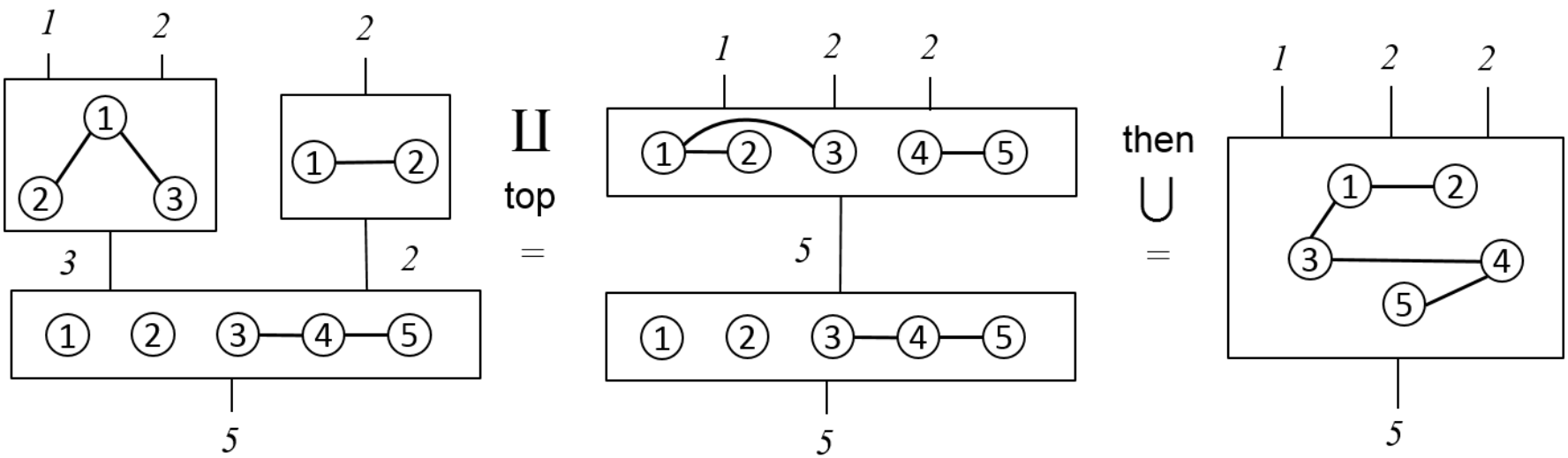
The key fact is that **networks serve as operations to assemble networks**, thanks to our ability to overlay them.

The simple network operad

Let $F = \mathrm{SG}$ be such that $\mathrm{SG}(n)$ is that set of **simple graphs** on n vertices with 'overlay' given by taking the union of edge sets.

Note that we can overlay a graph in $\mathrm{SG}(n)$ on the disjoint union of graphs with $n_1, \ldots, n_k$ vertices as long as $n_1 + \cdots n_k = n$.

For instance, $f \circ (g, h) \in \mathrm{O}_{\mathrm{SG}}(1, 2, 2; 5)$ is formed as follows:

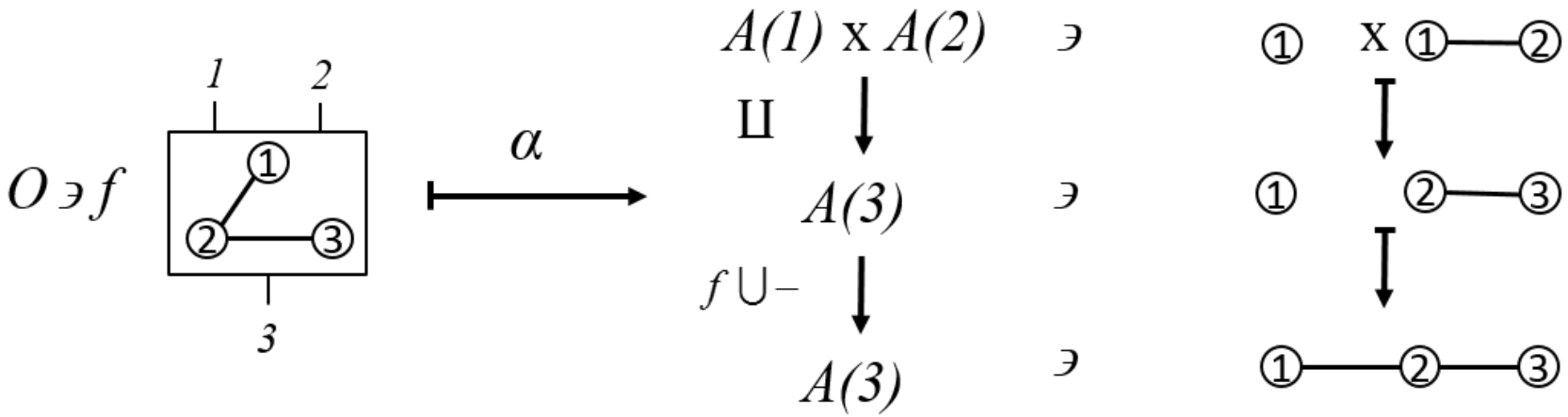


The canonical algebra: simple networks

We can use O_{SG} to act on simple graphs and call $A_{\mathrm{SG}}(n) = \mathrm{SG}(n)$ the 'canonical algebra' of O_{SG}.

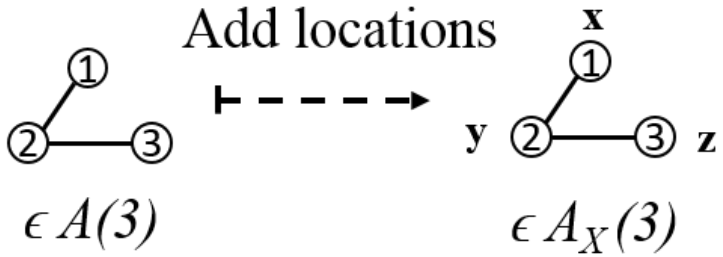
In A_{SG} each operation $(\sigma, g) \in S_n \times \mathrm{SG}(n)$ acts on a k-tuple of simple graphs h_i to give the simple graph

$$\alpha(\sigma, g)(h_1, \ldots, h_k) = g \cup \sigma(h_1 \sqcup \cdots \sqcup h_k) \in A_{\mathrm{SG}}(n).$$

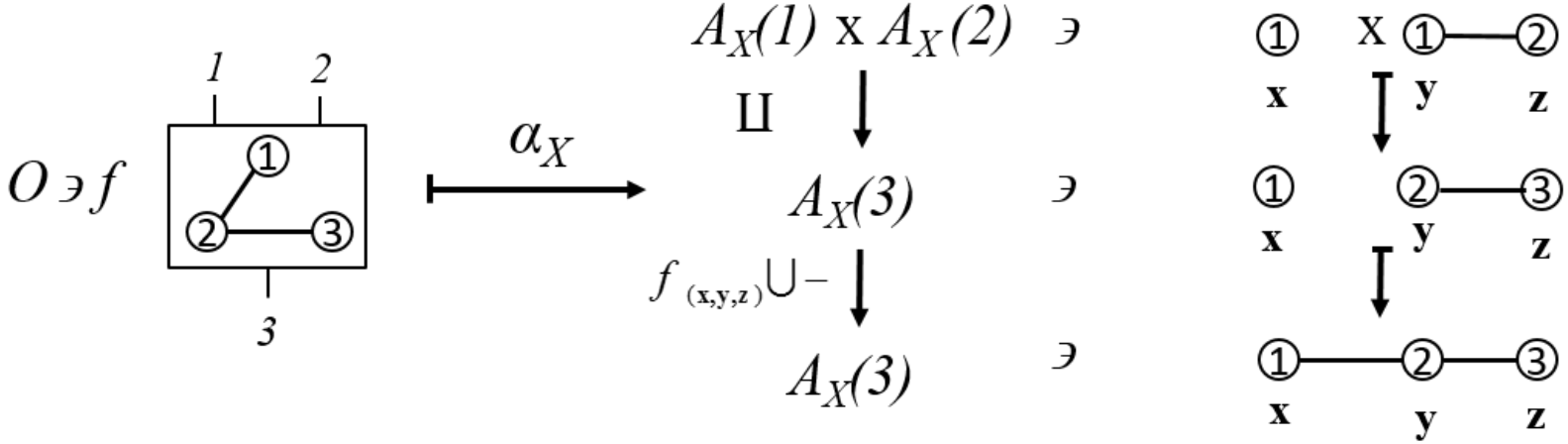


Simple networks with locations

O_{SG} can also act on simple graphs whose vertices have attributes in some set X. That is, $A_X(n) = \mathrm{SG}(n) \times X^n$.

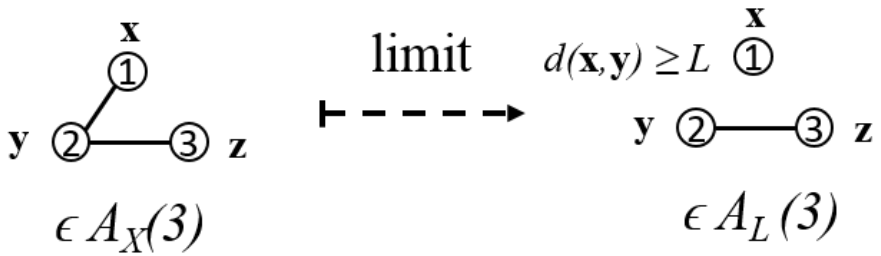


For example, if $X = \mathbb{R}^2$, we can view this as endowing vertices with a location in the plane.

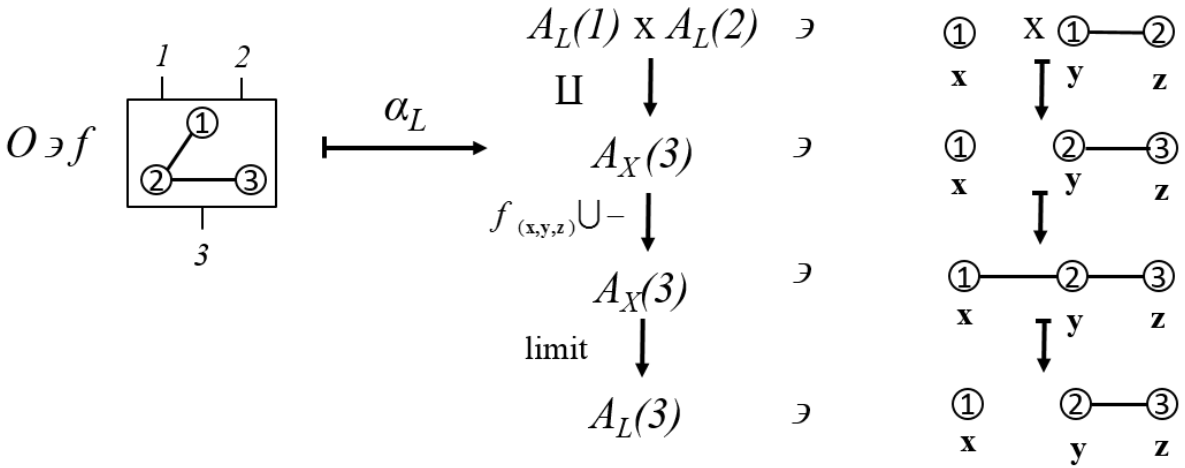


Range-limited networks

Now let $A_L(n) \subseteq \mathrm{SG}(n) \times \mathbb{R}^{2n}$ be those simple graphs with vertices in the plane such that no edge has length greater than L.



Enforcing this constraint is compatible with the O_{SG}-action and gives another O_{SG}-algebra:

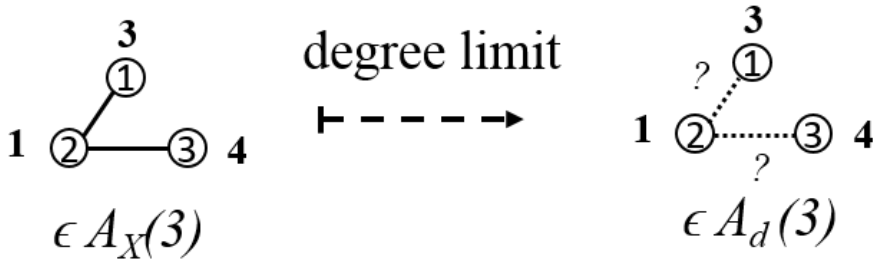


Degree-limited networks

Now suppose we want

- each vertex to have degree-limit in $\mathbb{N}$—e.g. p USB ports and
- to act on simple graphs $A_d(n) \subseteq \mathrm{SG}(n) \times \mathbb{N}^n$ such that no vertex has degree greater than it's limit.

Unlike range-limits, it's not obvious how to limit links: d



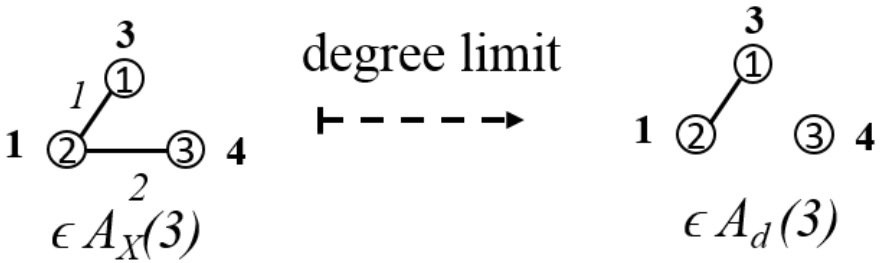
Basically, you're pushed to prioritize vertices (breaks symmetry) or remember how edges are added (non-commutative monoids).

Question

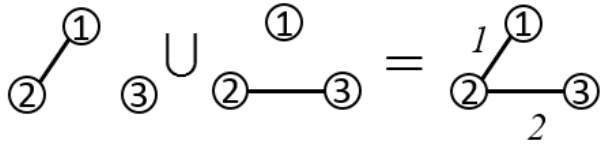
What is the network operad to compose degree-limited networks?

Degree-limited networks

Motivated by real-life we prioritize earlier operations—i.e. once a computer is full of USB cables later attempts to plug-in more cables fail—but the opposite convention is possible.



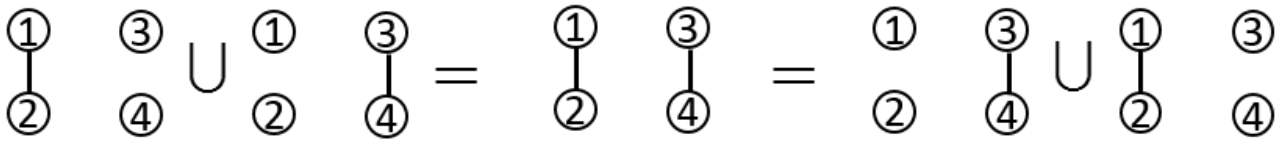
The figure captures that in the TBD monoid $F(3)$



but doesn't specify exactly how much we 'remember'.

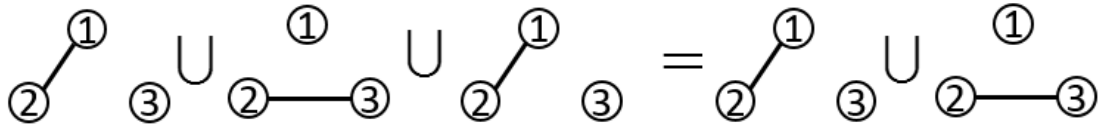
Graphic monoids

In fact, the interchange law demands some forgetfulness—e.g.



and we can demand the all graphs overlay as follows

$g \cup h \cup g = g \cup h$ e.g.



i.e. the second attempt link *the same devices* is superfluous.

- The graphic identity $aba = ab$ has been rediscovered many times and guarantees finitely generated $\implies$ finite.
- The name 'graphic' comes from Lawvere [5].
- Each $\mathrm{SG}(n)$ is a commutative graphic monoid.

An operad for degree-limited networks

Question

What is the network operad to compose degree-limited networks?

Answer

Graphic Identity + Interchange $\implies$ certain 'partially free' graphic monoids (with presentations like right-angled Artin groups) can be used to define $F\colon \mathbf{S} \to \mathbf{Mon}$ and such that A_d is a O_F-algebra.

Fun fact

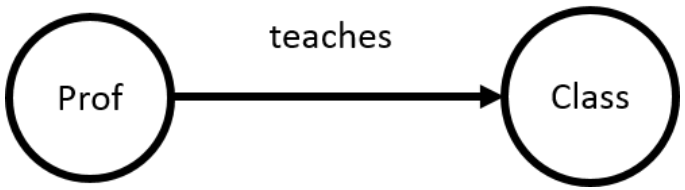
The graphic identity on a monoid is equivalent to the relation $ab = b \iff a \leq b$ defining a partially ordered set.

- The poset gives a direction for adding 'more' links.

Integrated design and tasking

Capacity limits—e.g. at most p USB cables—suggest that a non-commutative syntax to needed to *design* networks.

However, to *task* interactions between entities—e.g.



the agents' task commitments are inherently limited.

Right now, the state-of-the-art is to put in various limits on tasking syntax by hand [3].

So far, the graphic identity seems to

- balance tractability and expressiveness and
- capture that current design and tasking choices constrain future options.

Questions for ongoing and future work

- Which network models $F : \mathbf{S} \to \mathbf{Mon}$ extend to **Inj**, the category of finite sets and injective functions?
- Can we leverage the posets coming from the graphic identity to structure the search for 'good' designs and taskings?
- How can we systematically and iteratively construct a sequence of network operads and their algebras to gradually 'layer on' modeling complexity?
- Can functorial 'templates' for network models allow engineers to rapidly specify a network operad for a specific applications?

Further reading and acknowledgments

- Check our e-print: Network Models, arXiv:1711.00037.
- You out for postings of technical reports [2, 3] for more examples.

This work was supported by the DARPA Complex Adaptive System Composition and Design Environment (CASCADE) project.

We thank Chris Boner, Tony Falcone, Marisa Hughes, Joel Kurucar, Tom Mifflin, John Paschkewitz, Thy Tran and Didier Vergamini for many helpful discussions.

[1] J. C. Baez, J. D. Foley, J. Moeller and B. S. Pollard, Network Models, Preprint, Available as arXiv:1711.00037.

[2] J. C. Baez, J. D. Foley, J. Moeller and B. S. Pollard, Operads for communication networks, 2016, 37 pages, Technical report, DARPA CASCADE project.

[3] J. C. Baez, J. D. Foley, J. Moeller and B. S. Pollard, Compositional tasking, 2017, 27 pages, Technical report, DARPA CASCADE project.

[4] N. Gambino and A. Joyal, On operads, bimodules and analytic functors, *Memoirs AMS* **249**, 2017. Available as arXiv:1405.7270

[5] F. W. Lawvere, Qualitative distinctions between some toposes of generalized graphs, *Contemp. Math.* **92** (1989) pp. 261-299.

[6] D. Yau, *Colored Operads*, American Mathematical Society, Providence, Rhode Island, 2016.