**H.J. Sieburgh**

# Simplicial Coalgebras for Concurrent Regular Languages

**Bachelor thesis**

**August 15th, 2025**

**Thesis supervisors:** dr. H. Basold
dr. M. Hablicsek
dr. C. Ford

**Leiden University**
**Mathematical Institute**
**Leiden Institute of Advanced Computer Science**

## Abstract

Regular expressions give a~~n easy~~ way to express regular languages, and through Kleene's theorem we know the regular languages they describe are exactly equal to the ones modelled by deterministic finite automata. These automata give ~~a very~~ efficient way to parse strings, search through data, check for patterns etc. ~~But these~~ regular languages only contain strings of characters, given by sequential composition. In practice machines, software, biology, and other forms of data exhibit concurrent behaviour. Such as parallel computing, manufacturing with multiple steps which aren't directly dependent, or cellular activity in organisms.

In contrast to regular languages, a model for ~~these~~ concurrent languages, together with expressions, hasn't been found yet. Previous attempts have been made but they haven't been as useful as wanted. We think the main problem with the previous approaches have been the conflation of structure with behaviour, transitions in the model were restricted by the form of the state space.

For this problem we first propose a new type of dependency graph called ranked hypergraphs. These are directed acyclic hypergraphs which have root and output interfaces that give control over dependencies and the defined sequential composition is associative ~~up to isomorphism~~.

Because of the interfaces there is a generator of graphs of depth 1. Which means that in these graphs there isn't a path longer than 1 edge. We can inductively split up any graph into these pieces, which lets them serve as building blocks for all ranked hypergraphs.

What is the context for this sentence?

As structure for these graphs we define a simplicial set containing them. Finally, a coalgebra is used to define a transition and acceptance system for these states, mirroring classical automata by splitting up graphs into their building blocks.

# Contents

# 1 Introduction

Finite automata and regular languages form the foundation of classical formal language theory. Deterministic finite automata (DFA) provide a simple and elegant model for accepting regular languages, with Kleene's theorem stating their equivalence to regular expressions. In this setting, words are sequential compositions from single letters in an alphabet, and state transitions are deterministic.

However, many systems exhibit concurrent behaviour, where actions occur in parallel with complex dependencies. In a previous attempt to model this behaviour, higher-dimensional automata (HDA) were proposed. In HDA, concurrent actions are modelled using paths through hypercubes. But this combination of structure and behaviour is constricting and leads to just a small set of described languages. The missing part in particular is a parallel equivalent to the Kleene star operation.

In this thesis, we develop an approach to model concurrent regular languages which separates structure from behaviour: simplicial coalgebras over ranked hypergraphs.

We introduce ranked hypergraphs with interfaces that allow associative composition up to renaming. The interfaces give a way to connect dependencies. This composition allows for a generator which consists of graphs of depth 1.

The generator allows for piece-by-piece parsing of a graph. Mirroring how finite automata parse a word character by character. With the alphabet being the generator.

We will first introduce the necessary background on simplicial sets, subobject identifiers and coalgebras. We then formalize ranked hypergraphs and their compositions, describe the simplicial structure, and finally define the coalgebraic behaviour function which gives the parsing semantics.

## 1.1 Related Work

**Interface pomsets** In discrete finite automata (DFA) over an alphabet $\Sigma$ we work with single characters, which when sequentially composed build the words of languages. But in concurrent settings we need actions which have multiple events that can be parallel, dependent, weakly dependent, etcetera. But to be useful they also need a structure that allows composition, and most ideally a generator.

For this, interface pomsets were introduced [ABC+24].

**Definition 1.1.** [ABC+24] An *ipomset* (over an alphabet $\Sigma$) is a structure $(P, <, \dashrightarrow, S, T, \lambda)$ consisting of the following:

- a finite set $P$ of *events*,
- a strict partial order (i.e., asymmetric, transitive, irreflexive) $< \subseteq P \times P$ called the precedence order,
- a strict partial order $\dashrightarrow \subseteq P \times P$ called the *event order*,

- a subset $S \subseteq P$ called the *source set* of *sources*,
- a subset $T \subseteq P$ called the *target set* of *targets*,
- a *labelling* map $\lambda : P \to \Sigma$.

Intuitively we understand the precedence order, if $a < b$ then $a$ must end before $b$ begins. The event order is a weaker notion which is necessary for sequential composition. The source and target sets are also used for composition, we can only *glue* two ipomsets $P$ and $Q$ together if $T_P \cong S_Q$.

We identify a couple of important types of ipomsets. Namely, an ipomset $(P, <, \dashrightarrow, S, T, \lambda)$ is:

- *discrete* if $<$ is empty,
- a *conclist* (short for 'concurrency list') if it is discrete and $S = T = \varnothing$,
- a *starter* if it is discrete and $T = P$,
- a *terminator* if it is discrete and $S = P$, and
- an *identity* if it is both a starter and a terminator.

An *isomorphism* of ipomsets $P$ and $Q$ is a bijection $f : P \to Q$ such that:

1. $f(S_P) = S_Q$, $f(T_P) = T_Q$, $\lambda_Q \circ f = \lambda_P$,
2. $f(x) <_Q f(y) \iff x <_P y$, and
3. $x \not<_P y$ and $y \not<_P x$ imply that $x \dashrightarrow_P y \iff f(x) \dashrightarrow_Q f(y)$.

**Proposition 1.2.** Isomorphism classes of ipomsets form a category $\mathrm{iiPoms}_{\simeq}$.

- objects are isomorphism classes of conclists,
- morphisms in $\mathrm{iiPoms}_{\simeq}(U, V)$ are isomorphism classes of ipomsets $P$ with $S_P = U$ and $T_P = V$,
- composition of morphisms is gluing, and
- identities are identity ipomsets, i.e. $(U, \varnothing, \dashrightarrow, U, U, \lambda) \in \mathrm{iiPoms}_{\simeq}(U, U)$.

Finally, let $\bar{\Omega}_{\simeq}$ be the directed multigraph given as follows:

- Vertices are objects of $\mathrm{iiPoms}_{\simeq}$.
- Edges in $\bar{\Omega}_{\simeq}$ are morphisms in $\mathrm{iiPoms}_{\simeq}(U, V)$

**Corollary 1.3** ([ABC+24]). The category $\mathrm{iiPoms}_{\simeq}$ is generated by the directed multigraph $\bar{\Omega}_{\simeq}$ using gluing composition under the identities [ABC+24, equation. (2)].

The concept of discreteness in starters and terminators inspired the generator of ranked hypergraphs, described in Theorem 3.12. In our case, if an edge has a predecessor then the action of that predecessor has to happen first. No precedence order would therefore translate to all edges having no predecessors. Therefore giving directed graphs of depth 1.

**Higher Dimensional Automata** Higher Dimensional Automata (HDA) have been proposed as an analogue of DFA for modelling concurrent languages [Pra91, Gla91]. An HDA is a *precubical set* with start and end points. Which means its states have the structure of hypercubes, with direction over its edges inducing direction for cells of all dimensions.

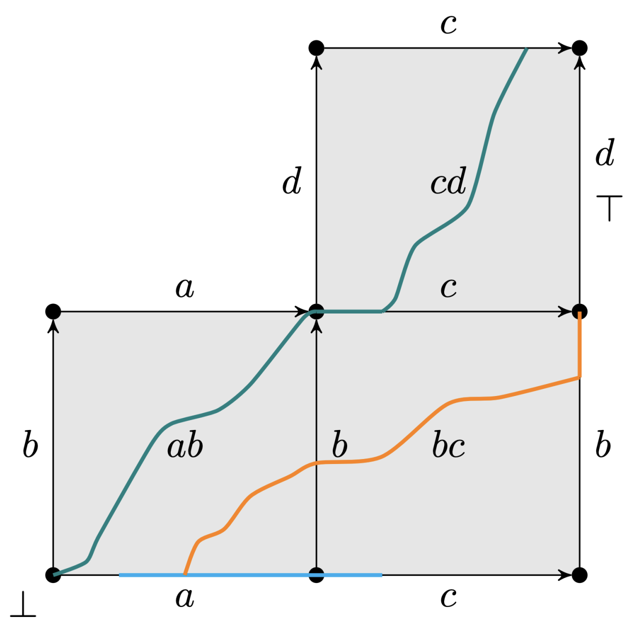


Figure 1: Paths in a HDA

A Kleene theorem was found for these automata by Fahrenberg et. al. [FJSZ24] which proves the languages HDA define are exactly the rational subsumption-closed sets of finite interval pomsets. One of the limitations of this is that the rational operations used by Fahrenberg et. al. don't include a 'parallel Kleene star' but only a binary (therefore finite) parallel composition. Thus severely restricting the description of concurrency by this class of languages.

# 2 Background

## 2.1 Preliminaries

This thesis assumes basic knowledge of and experience with automata theory and category theory. A good starting point for automata would be chapters 2-3 of *Introduction to Languages and the Theory of Computation* by John C. Martin [Mar11]. For category theory I would recommend the classic book *Categories for the Working Mathematician* [Lan98].

## 2.2 Semilattices

**Definition 2.1.** Let $P$ be a partially ordered set (poset), the *join* (or least upper bound) is a binary operator $\vee$ such that for all $x, y, z \in P$:

- $x \leq x \vee y$,
- if $x \leq a$ and $y \leq a$ then $x \vee y \leq a$.

the *meet* (or greatest lower bound) is a binary operator $\wedge$ such that for all $x, y, z \in P$:

- $x \wedge y \leq x$,
- if $z \leq x$ and $z \leq y$ then $z \leq x \wedge y$.

**Definition 2.2.** A *join-semilattice* is a triple $(A, \vee, \perp)$ consisting of

- A partially ordered set $A$,
- A binary join $\vee : A \times A \to A$ which is commutative, associative, and idempotent: $a \vee a = a$,
- A unit $\perp$ such that $a \vee \perp = a$.

The collection of all join-semilattices forms a category JSL.

A *meet-semilattice* is the same as a join-semilattice, except we have as operator a meet instead of a join.

An object that is both a meet-, and a join-semilattice is called a *lattice*.

**Definition 2.3.** Let $X$ be a set. Let $\mathcal{P}_\omega(X)$ be the set of finite subsets of $X$, then $(\mathcal{P}_\omega(X), \cup, \varnothing)$ is a JSL.

**Definition 2.4.** The free join-semilattice on a set $X$ is the unique JSL $F(X)$ together with a function $\eta : X \to F(X)$ such that for all $A \in \text{JSL}$, $f : X \to A$ there exists a unique JSL morphism $\hat{f} : F(X) \to A$ such that the following diagram commutes:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\eta"] & A \\ F(X) \arrow[ur, "\hat{f}"'] & \end{tikzcd}$$

This definition is a special case of the more general notion of a free object [AHS09].

**Lemma 2.5.** Let $X \in \mathbf{Set}$, $\mathcal{P}_\omega(X)$ is the free JSL on $X$.

*Proof.* Take $\eta(x) = \{x\}$. Let $A$ be a join-semilattice, $f : X \to A$ a function and let $\hat{f}(\{x_1, \ldots, x_n\}) = f(x_1) \vee \cdots \vee f(x_n)$.

Then $(\hat{f} \circ \eta)(x) = \hat{f}(\{x\}) = f(x)$. And thus the finite powerset is the unique free JSL. $\square$

**Definition 2.6.** A Heyting algebra H is a lattice together with the operation of implication $\Rightarrow: H \times H \to H$ which for all $x, y, z \in H$ satisfies:

$$x \wedge y \leq z \iff x \leq y \Rightarrow z$$

Moreover, a Heyting algebra where the *principle of excluded middle* holds, which states every proposition is either true or false, is a Boolean algebra. When writing $\vdash$ for $\leq$ you get precisely the classical notion of propositional logic.

## 2.3 Presheaves

**Definition 2.7.** Let $C$,$D$ be categories, and $F, G : C \to D$ be functors. Then a *natural transformation* $\mu : F \Rightarrow G$ from $F$ to $G$ is a family of morphisms such that:

1. for all $x \in C$ there exists a morphism $\mu_x : F(x) \to G(x)$ called the component at $x$
2. for all $x, y \in C$ and all morphisms $f : x \to y$ the following diagram commutes:

$$\begin{tikzcd} F(x) \arrow[r, "F(f)"] \arrow[d, "\mu_x"] & F(y) \arrow[d, "\mu_y"] \\ G(x) \arrow[r, "G(f)"] & G(y) \end{tikzcd} \tag{1}$$

**Definition 2.8.** Let $C$ be a category, a *presheaf* is a functor

$$F : C^{op} \to \mathbf{Set}$$

The collection of all presheaves on a category forms a category with as morphisms the natural transformations between the presheaves.

## 2.4 Simplicial sets

**Definition 2.9.** The simplex category $\Delta$ has as objects

$$[n] = \{0 < 1 < \cdots < n\}$$

and as morphisms the order-preserving functions between them.

For all $n \in \mathbb{N}$, $i \in [n]$ we define the coface map $\delta_i^n : [n-1] \to [n]$ as the injective order preserving map which has $(\delta_i^n)^{-1}(i) = \varnothing$. So the map 'misses' $i$ and all other elements of $[n]$ are 'hit'.

For all $n \in \mathbb{N}$, $i \in [n]$ we define the coface map $\sigma_i^n : [n-1] \to [n]$ as the surjective order preserving map which has $|(\sigma_i^n)^{-1}(i)| = 2$, $|(\sigma_i^n)^{-1}(j)| = 1 \ \forall j \neq i$. So the map 'hits' $i$ twice, and all other $j$ once.

All morphisms of $\Delta$ are generated from the coface and codegeneracy maps.

*Notation:* $\delta_i^n$, $\sigma_i^n$ and so on, are just referred to as $\delta_i$, $\sigma_i$ etc. As the exact $n$ is not necessary for most applications.

**Definition 2.10.** A *simplicial set* is a presheaf on $\Delta$, which means a simplicial set is a functor

$$X \colon \Delta^{\mathrm{op}} \to \mathbf{Set}$$

Thus, a simplicial set $X$ assigns to each $[n] \in \Delta$ a set $X_n = X([n])$ of $n$*-simplices*, and to each morphism $\theta \colon [m] \to [n]$, a function $X(\theta) \colon X_n \to X_m$.

In particular, we have $X(\delta_i) = d_i$ where $d_i : X_n \to X_{n-1}$ omits the $i$-th element from an $n$-simplex. And $X(\sigma_i) = s_i$ where $s_i : X_n \to X_{n+1}$ repeats the $i$-th element in the simplex.

These satisfy the *simplicial identities*:

$$\begin{aligned}
d_i d_j &= d_{j-1} d_i && \text{if } i < j, \\
s_i s_j &= s_{j+1} s_i && \text{if } i \leq j, \\
d_i s_j &= \begin{cases} s_{j-1} d_i & \text{if } i < j, \\ \mathrm{id} & \text{if } i = j \text{ or } i = j+1, \\ s_j d_{i-1} & \text{if } i > j+1. \end{cases}
\end{aligned}$$

The presheaf category of simplicial sets is called sSet and its natural transformations the *simplicial morphisms*.

Given a collection of sets $S = S_0 \sqcup S_1 \sqcup \ldots$ and functions $d_i : S_n \to S_{n-1}$, $s_i : S_n \to S_{n+1}$ satisfying the simplicial identities there is a unique simplicial set which has the same face and degeneracy maps.

**Lemma 2.11.** Let $X, Y \in \mathrm{sSet}$, a collection of morphisms $f_0 : X_0 \to Y_0, f_1 : X_1 \to Y_1, \ldots$ is a simplicial morphism if and only if the following diagrams commute.

$$\begin{tikzcd}
X_n \arrow[r, "d_i^X"] \arrow[d, "f_n"] & X_{n-1} \arrow[d, "f_{n-1}"] \\
Y_n \arrow[r, "d_i^Y"] & Y_{n-1}
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "s_j^X"] \arrow[d, "f_n"] & X_{n+1} \arrow[d, "f_{n+1}"] \\
Y_n \arrow[r, "s_j^Y"] & Y_{n+1}
\end{tikzcd}$$

*Proof.* Given that $\delta_i$, $\sigma_i$ generate the morphisms in $\Delta$, diagram 1 automatically commutes for all $f$ if it does so for $\delta$ and $\sigma$. $\square$

**Definition 2.12.** Let $U$ be a poset, denote by $U^{\triangleright}$ the poset $U$ with a new maximal element added. On $\Delta$ this gives $[n]^{\triangleright} \cong [n+1]$.

This gives a functor $\uparrow : \mathrm{sSet} \to \mathrm{sSet}$ by $\uparrow X = X \circ (-)^{\triangleright}$. Then for $x \in X_n$ and $f : X \to \uparrow X$ we have $f(x) \in X_{n+1}$.

## 2.5 Subobject Classifiers

**Definition 2.13.** Let $f, g$ be morphisms defined by

$$\begin{tikzcd}
& Y \arrow[d, "g"] \\
X \arrow[r, "f"'] & Z
\end{tikzcd}$$

Then a *pullback* is an object $P$ together with morphisms $p_1, p_2$ such that

$$\begin{tikzcd}
P \arrow[r, "p_2"] \arrow[d, "p_1"] & Y \arrow[d, "g"] \\
X \arrow[r, "f"'] & Z
\end{tikzcd}$$

commutes and is universal. The universal property means that for any other object $Q$, with morphisms $q_1 : Q \to X$, $q_2 : Q \to Y$ such that they commute just like the above, there needs to be a morphism $\beta : Q \to P$ such that this whole diagram commutes.

$$\begin{tikzcd}
Q \arrow[dr, dashed, "\beta"] \arrow[drr, bend left, "q_2"] \arrow[ddr, bend right, "q_1"'] & & \\
& P \arrow[r, "p_2"] \arrow[d, "p_1"] & Y \arrow[d, "g"] \\
& X \arrow[r, "f"'] & Z
\end{tikzcd}$$

**Definition 2.14.** Let $C$ be a category with a terminal object $*$ and finite limits. A *subobject* *classifier* $\Omega$ is an object in $C$ such that there exists a morphism $\top : * \to \Omega$ which has the property that for all objects $A, X$ and monomorphisms $f : A \to X$ there exists a unique morphism $\chi_A : X \to \Omega$ that makes the diagram commute and be a pullback

$$\begin{tikzcd} A \arrow[r] \arrow[d, "f"] & * \arrow[d, "\top"] \\ X \arrow[r, "\chi_A"] & \Omega \end{tikzcd}$$

**Example 2.15.** In the category Set where objects are sets and morphisms are functions, the subobject classifier is $\Omega = \mathbb{B} := \{0, 1\}$ and if $f = \mathrm{id}$ then $\chi_A$ is the characteristic map of the subset $A \subseteq X$.

**Lemma 2.16.** sSet has a subobject classifier $\Omega$ which has an internal Heyting algebra.

*Proof.* From [Joh02, Lemma. 1.6.6] and [Joh02, Lemma. 1.6.3.ii]. □

## 2.6 Coalgebras

In this thesis we primarily work with $F$-coalgebras, which we will introduce next, these are a special type of coalgebra which fits our use-case very well. A general definition therefore is not necessary as the definition of $F$-coalgebras is self-contained.

**Definition 2.17.** Given a category $C$ and an endofunctor $F : C \to C$. An $F$-coalgebra $(A, \alpha)$ consists of an object $A \in C$ called the carrier and a morphism $\alpha : A \to FA$.

**Example 2.18** (Labelled transition systems)**.** Take as carrier a state space $X$, and a set of actions (or characters in the context of DFA) $A$.

Take as endofunctor

$$FX = \mathcal{P}(A \times X)$$

Then given a state $x \in X$, $\alpha(x) = \{(a_1, x_1), (a_2, x_2), \dots\}$ denotes possible transitions to states $x_1, x_2, \dots$ with corresponding outputs of $a_1, a_2, \dots$.

In figure 2.18 a labelled transition system is given where $\alpha(1) = \{(a, 3), (b, 2)\}$, $\alpha(2) = \{(a, 3)\}$, $\alpha(3) = \varnothing$.

**Definition 2.19.** A pointed $F$-coalgebra over a functor $F$ is a triple $(X, \alpha : X \to FX, x_0)$ where $X$ is the carrier as usual and $x_0$ is the *base* or in our case an *initial state*.

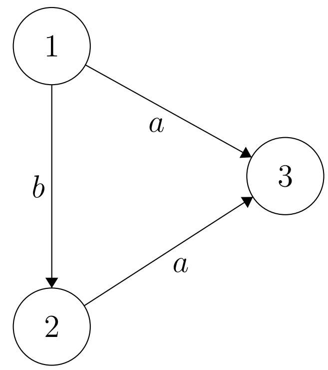


Figure 2: A labelled transition system

# 3 Ranked Hypergraphs

**Definition 3.1.** A directed *hypergraph* $(V, H)$ is a finite set of vertices $V$ and a set of *hyperarcs* $H \subseteq V \times \mathcal{P}(V)$.

*Notation:* A directed hypergraph containing no cycles is a Directed Acyclic Hypergraph (DAH).

**Definition 3.2.** A *ranked hypergraph* $(g, r, o, \mathcal{L}, A)$ consists of:

- A DAH $g$,
- Finite sequences $r = (r_i)_{i \leq |r|}$, $o = (o_i)_{i \leq |o|}$ $r_i, o_i \in \mathcal{P}(V)$ denoting the root and variable interfaces. $o_i$ contains only maximal vertices. We refer to $(|r|, |o|)$ as the *rank* of this graph.
- An action set $A$ and a hyperarc labelling function $\mathcal{L} : H \to A$

*Notation:* In this thesis we often refer to ranked hypergraphs as just hypergraphs as we will only be working with this kind. $HG(n, m)$ is the set of ranked term hypergraphs of rank $(n, m)$

**Example 3.3.** In figure Example 3 a ranked hypergraph is drawn. Left is the root interface, of rank 3, on the right is the output interface of rank 2.

The full definition of this graph is as follows

- $V = \{1, 2, 3, 4, 5, 6\}$,
- $H = \{(1, \{4\}), (2, \{5, 6\})\}$,
- $r = (\{1\}, \{2\}, \{3\})$, $o = (\{4\}, \{3, 6\})$,

- $A = \{a, b\}$, $\mathcal{L}((1, \{4\})) = b$, $\mathcal{L}((2, \{5, 6\})) = a$.

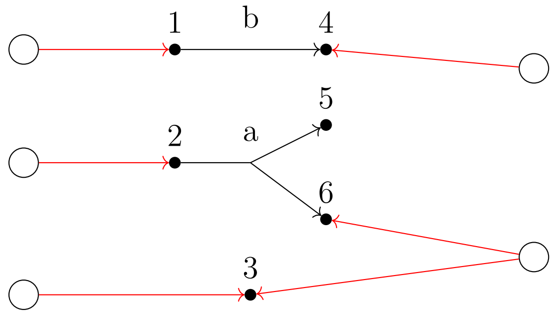


Figure 3: A Directed Acyclic Hypergraph

**Definition 3.4.** Let $G, H \in HG(n, m)$ be ranked hypergraphs of equal rank $(n, m)$,

An *isomorphism* between $G$ and $F$ is a bijection

$$f : V^G \to V^F$$

such that

$$\begin{aligned}(u, U') \in H^G &\iff (f(u), f[U']) \in H^F \\ v \in r_i^G &\iff f(v) \in r_i^F \quad \forall i \leq n \\ v \in o_i^G &\iff f(v) \in o_i^F \quad \forall i \leq m\end{aligned}$$

**Remark 3.5.** When proving a property which involves multiple hypergraphs, we can assume disjointness of vertex sets. But then the property will only hold up to isomorphism. This is because renaming of vertices is an isomorphism in the sense of Definition 3.4.

## 3.1 Composition of ranked hypergraphs

**Definition 3.6.** Let $G, F$ be hypergraphs such that $|o^G| = |r^F|$, their composition is defined as follows:

$$G; F = (g', r', i_F[o^F] := (i_F[o_i^F])_{i \leq |o^F|}, \mathcal{L}^G \sqcup \mathcal{L}^F, A^G \cup A^F) \tag{2}$$

We obtain $g' = (V, H)$ by the following procedure:

Define

$$V = (V^G + V^F) \setminus i_G[\bigcup_{i \leq |o^G|} o_i^G]$$

To get the hyperarcs we keep all elements but replace a vertex if it exists in an output, to do this neatly we define a pair of functions:

$$\psi_{G,F}(v) := \begin{cases} i_F[\bigcup\limits_{\substack{i \leq |o^G| \\ v \in o_i^G}} r_i^F] & \text{if } \exists i \leq |o^G| \text{ such that } v \in o_i^G \\ \{i_G(v)\} & \text{else} \end{cases}$$

$$\Psi_{G,F}(U') := \bigcup_{v \in U'} \psi_{G,F}(v)$$

$$H := \left\{(i_G(u), \Psi_{G,F}(U')) : (u, U') \in H^G\right\} \cup i_F[H^F]$$

So for all $i$, in each arc $v$ that ends in a vertex in $o_i^G$, we replace that vertex in the arc with $r_i^F$.

And we obtain $r'$ by taking over the original $r^G$ and 'connecting through' for vertices which are both minimal and maximal:

$$r'_i = \Psi_{G,F}(r_i^G)$$

**Remark 3.7.** When vertex sets are disjoint, we can simply take set union as coproduct. Which makes the inclusions identity maps.

This composition allows for a left identity $id_n$ namely $id_n = ((\{1, 2, \ldots, n\}, \varnothing), (\{i\})_{i \in n}, (\{i\})_{i \in n})$.

**Example 3.8.** Figure 4 shows the sequential composition $G; F$. In this example it is visible that interface points give a handle on dependencies. For instance, as $4 \in o_G$, and $7 \in r_F$, in the final product 7 is dependent on action $b$,

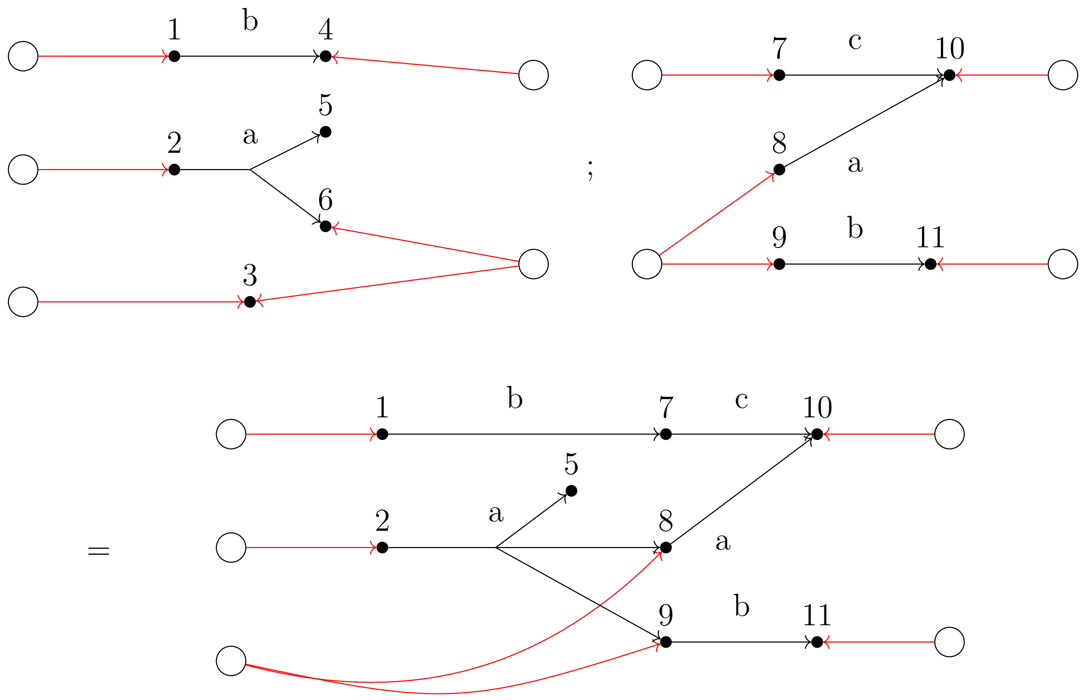


Figure 4: The composition of two ranked hypergraphs

**Lemma 3.9.** Let $G, K, F$ be DAH and $U \subseteq V^G$ a set of vertices, it holds that:

$$\Psi_{G;K,F} \circ \Psi_{G,K}(U) \cong \Psi_{G,F;K}(U)$$

*Proof.* Let $G, K, F$ be DAH and $U \subseteq V^G$ a set of vertices. Assume disjoint vertex sets.

We prove the lemma for a singleton, this then extends trivially to all subsets $U \subseteq V^G$. Let $v \in V^G$, we proceed by using the definitions:

$$\Psi_{G;K,F} \circ \psi_{G,K}(v) = \begin{cases} \{v\} & v \notin \cup_i o_i^G \\ \Psi_{G;K,F}\left(\bigcup_{\substack{j \leq |r^K| \\ v \in o_j^G}} r_j^K\right) & \text{else} \end{cases}$$

Because both unions are finite we can exchange them:

$$= \begin{cases} \{v\} & v \notin \cup_i o_i^G \\ \bigcup_{\substack{j \leq |r^K| \\ v \in o_j^G}} \Psi_{G;K,F}(r_j^K) & \text{else} \end{cases}$$

By definition $r_i^{K;F} := \Psi_{K,F}(r_i^K)$ and hence

$$= \begin{cases} \{v\} & v \notin \cup_i o_i^G \\ \bigcup_{\substack{j \leq |r^{K;F}| \\ v \in o_j^G}} r_j^{K;F} & \text{else} \end{cases}$$

$$= \psi_{G,K;F}(v)$$

Taking unions of singletons then yields:

$$\Psi_{K,F} \circ \Psi_{G,K}(U) = \Psi_{G,K;F}(U)$$

By Remark 3.5 the equality holds up to isomorphism for all $G, K, F$, regardless of vertex sets. $\square$

**Theorem 3.10.** Let G,K,F be ranked hypergraphs. The following properties hold:

1. $G; (K; F) \cong (G; K); F$
2. $id_{|r^G|}; G \cong G$

*Proof.* $G; (K; F) \cong (G; K); F$
Let $G, K, F$ be ranked hypergraphs with disjoint vertex sets. Use Remark 3.7. It is clear from the definition that $(G; K); F = G; (K; F)$ if and only if their graphs and root interfaces are equal.

Let $r$, $r'$ be the root interfaces for $(G; K); F$, $G; (K; F)$ respectively. We expand the definition and use Lemma 3.9 to show that $r = r'$. Let $i \leq |r^G|$, we get:

$$r_i = \Psi_{r^F,o^k}(r_i^{G;K}) = \Psi_{r^F,o^k}(\Psi_{r^K,o^G}(r_i^G)) = \Psi_{r^{K;F},o^G}(r_i) = r_i'$$

Let $g = (V, H)$, $g' = (V', H')$ be the graphs for $(G; K); F$, $G; (K; F)$ respectively.

We expand the definition:

$$\begin{aligned} V &= (V^{G;K} + V^F) \setminus \bigcup_{i \leq |o^K|} o_i^K \\ &= (((V^G + V^K) \setminus \bigcup_{i \leq |o^G|} o_i^G) + V^F) \setminus \bigcup_{i \leq |o^K|} o_i^K \end{aligned}$$

From disjointness of $V^G$, $V^K$, and $V^F$ we get

$$= (V^G + ((V^K + V^F) \setminus \bigcup_{i \leq |o^K|} o_i^K)) \setminus \bigcup_{i \leq |o^G|} o_i^G = V'$$

Lastly for the arcs:

$$\begin{aligned} H &= \{(u, \Psi_{r^F,o^K}(U')) : (u, U') \in H^{G;K}\} \cup H^F \\ &= \{(u, \Psi_{r^F,o^K}(\Psi_{r^K,o^G}(U'))) : (u, U') \in H^G\} \cup \{(u, \Psi_{r^F,o^K}(U')) : (u, U') \in H^K\} \cup H^F \end{aligned}$$

By Lemma 3.9 we have

$$\begin{aligned}
&= \left\{\left(u, \Psi_{r^{F;K}, o^G}(U')\right) : (u, U') \in H^G\right\} \cup \left\{\left(u, \Psi_{r^F, o^K}(U')\right) : (u, U') \in H^K\right\} \cup H^F \\
&= \left\{\left(u, \Psi_{r^{K;F}, o^G}(U')\right) : (u, U') \in H^G\right\} \cup H^{K;F} \\
&= H'
\end{aligned}$$

Which proves equality for disjoint graphs. Which in turn gives $G;(K;F) \cong (G;K);F$ for any $G, K, F$ by using Remark 3.5. □

*Proof.* $id_n; G \cong G$
Let $G$ be a ranked hypergraph with $|r^G| = n$ and

$$id_n = ((\{1, 2, \ldots, n\}, \varnothing), (\{i\})_{i\in n}, (\{i\})_{i\in n})$$

Assume disjoint vertex sets. Using remark 3.7 we get:

$$V = (\{1, 2, \ldots, n\} \cup V^G) \setminus \bigcup_{i\in n} \{i\} = V^G$$

$$H = \left\{(u, \Psi_{id_n;G}(U')) : (u, U') \in H^{id_n}\right\} \cup H^G = H^G$$

$$\psi_{id_n;G}(i) = r_i^G \quad \forall i \in \{1, 2, \ldots, n\}$$

Therefore

$$r'_i = \Psi_{id_n;G}(r_i^{id_n}) = \psi_{id_n;G}(i) = r_i^G$$

From 3.5 we conclude $id_n; G \cong G$ for any ranked hypergraph $G$. □

## 3.2 Generating hypergraphs

**Definition 3.11.** First define the notion of predecessors of a vertex as:

$$\text{pred}(v) = \{w \in V : \exists (w, U') \text{ s.t. } v \in U'\}$$

Then we can define the depth of a vertex recursively

$$d : V \to \mathbb{N}, \quad d(v) = \begin{cases} \max\{d(w) + 1 : w \in \text{pred}(v)\} & \text{pred}(v) \neq \varnothing \\ 0 & \textit{else} \end{cases}$$

which extends to a notion of depth for an entire ranked hypergraph:

$$D(G) = \max\{d(v) : v \in V^G\}$$

**Theorem 3.12.** The set of ranked hypergraphs is generated by the subset

$$HG_1 := \{G \in HG : D(G) \leq 1\}$$

That is, for all $G \in HG$ there exists an $n \in \mathbb{N}_{\geq 1}$ and a sequence $(G_1, G_2, \ldots, G_n) \in HG_1^n$, such that $G_1; G_2; \ldots; G_n \cong G$

*Proof.* To prove this by induction we first need the following result:
*Claim:* for any hypergraph $G$ of depth $n \geq 2$ there exist hypergraphs $G_1 \in HG$ with $D(G_1) = n-1$, $G_2 \in HG_1$ such that $G \cong K; F$.

First take $r^1 = r$, $o^2 = o$, $\mathcal{L}^1 = \mathcal{L}^2 = \mathcal{L}$, $A^1 = A^2 = A$.

Then let

$$\begin{aligned}
V^1 &= \{v \in V : d(v) \leq n-1 \text{ or } \exists w \in \text{pred}(v) \text{ s.t. } d(w) < n-1\} \\
V^2 &= \{v \in V : n-1 \leq d(v) \leq n\} \cup \bigcup_{i \leq |o^G|} o_i^G
\end{aligned}$$

$$\begin{aligned}
H^1 &= \{(u, U') \in H : d(u) < n-1\} \\
H^2 &= \{(u, U') \in H : d(u) = n-1\}
\end{aligned}$$

Note that for all $(u, U') \in H_1$ we have $u \in V^1 \setminus V^2$ because its degree is $< n-1$ and it isn't maximal and therefore not part of $o$. Furthermore, $(H^1, H^2)$ partitions $H$.

Finally let $o^1 = r^2$ be any enumeration of $V^1 \cap V^2$.

Let $i_1 : V^1 \to V^1 + V^2$, $i_2 : V^2 \to V^1 + V^2$ be the coproduct inclusions. We prove by construction of $G_1; G_2$ that the following function is an isomorphism

$$f : V \to (V^1 + V^2) \setminus i_1[\bigcup_{i \leq |o^1|} o_i^1], \quad f(v) = \begin{cases} i_1(v) & v \in V^1 \setminus V^2 \\ i_2(v) & \text{else} \end{cases}$$

First work out

$$\psi_{r^2,o^1}(v) := \begin{cases} i_2[\bigcup_{\substack{i \leq |o^1| \\ v \in o_i^1}} r_i^2] & \text{if } \exists i \leq |o^1| \text{ such that } v \in o_i^1 \\ \{i_1(v)\} & \text{otherwise} \end{cases}$$

by $o^1 = r^2$ we get

$$\begin{aligned}
&= \begin{cases} \{i_2(v)\} & \text{if } v \in V^1 \cap V^2 \\ \{i_1(v)\} & \text{otherwise} \end{cases} \\
&= \{f(v)\} \quad \forall v \in V^1
\end{aligned}$$

Therefore

$$\begin{aligned}
H^{G_1;G_2} &:= \left\{(i_1(u), f[U']) : (u, U') \in H^1\right\} \cup i_2[H^2] \\
&= \left\{(f(u), f[U']) : (u, U') \in H^1\right\} \cup \{(f(u), f[U'] : (u, U') \in H^2)\} \\
&= \{(f(u), f[U'] : (u, U') \in H)\}
\end{aligned}$$

and $r^{G_1;G_2} = (f[r_i^1])_{i \leq |r|} = (f[r_i])_{i \leq |r|}$, $o^{G_1;G_2} = (i_2[o_i^2])_{i \leq |o^F|} = (f[o_i])_{i \leq |o|}$

Which proves $f$ to be an isomorphism. Hence we get $G_1; G_2 \cong G$ as claimed, furthermore from the chosen $H^1$ we obtain $D(G_1) = n - 1$, $D(G_2) = 1 \implies G_2 \in HG_1$.

Now, let $G \in HG$ and assume $D(G) \leq 1$. This is trivially generated by $HG_1$ as $G \in HG_1$.

Proceed by induction, assume all $G \in HG$ with $D(G) = n$ are generated by $HG_1$. Take $G \in HG$ with $D(G) = n+1$. From the just proven result there exist $G_1 \in HG$, $G_2 \in HG_1$ such that $D(G_1) = n$. Therefore there exists a sequence $(G_{11}, \ldots, G_{1n}) \in$ such that $G_{11}; \ldots; G_{1n} \cong G_1$. Take as sequence $(G_{11}, \ldots, G_{1n}, G_2)$, then $G_1 1; \ldots; G_{1n}; G_2 \cong G_1; G_2 \cong G$. Thus all ranked hypergraphs of depth $n + 1$ are generated by $HG$.

By induction the statement holds for all $n$. Therefore $HG$ is generated by $HG_1$. $\square$

# 4 Simplicial sets over ranked term graphs

From this point on, the necessary structures are less certain. We would like to encode the concurrency into the dimensions of our state space, which makes behaviour more predictable as we can control when and where changes in dimension happen. There are multiple ways to do this, one of them would be to keep track of the amount of interfaces. This gives a handle on the amount of possible dependencies, composition etcetera, this is the approach chosen for Definition 4.3. Another, which is easier to parse semantically would be to keep track of the depth of the graph.

**Definition 4.1.** Let $V$ be the vertex set of a ranked hypergraph.
We define the monoid $\mathcal{M} = (\mathcal{P}(V)^2, (\varnothing, \varnothing), \cup \times \cup)$.

From this monoid we define a simplicial set using the nerve construction.

**Definition 4.2.** The *nerve* $N(\mathcal{M})$ of the monoid $\mathcal{M}$ is the simplicial set where:

$$N(\mathcal{M})_n = \mathcal{M}^n$$

$$d_i^{\mathcal{M}}(m_1, \dots, m_n) = \begin{cases} (m_1, \dots, m_i \cup \times \cup\, m_{i+1}, \dots, m_n) & 0 < i < n \\ (m_2, \dots, m_n) & i = 0 \\ (m_1, \dots, m_{n-1}) & i = n \end{cases}$$

$$s_i^{\mathcal{M}}(m_1, \dots, m_n) = (m_1, \dots, m_i, (\varnothing, \varnothing), m_{i+1}, \dots, m_n)$$

**Definition 4.3.** Define the simplicial set $\mathcal{H}$ by $\mathcal{H}_n = HG(n, n)$. With face and degeneracy maps:

$$\begin{aligned} &d_i^{\mathcal{H}}((g, (U_1, \dots, U_n), (W_1, W_2, \dots, W_n), \mathcal{L}, A)) \\ &\quad = \begin{cases} (g, (U_1, \dots, U_i \cup U_{i+1}, \dots U_n), (W_1, \dots, W_i \cup W_{i+1}, \dots W_n), \mathcal{L}, A) & 0 < i < n \\ (g, (U_2, \dots, U_n), (W_1, W_2, \dots, W_n), \mathcal{L}, A) & i = 0 \\ (g, (U_1, \dots, U_{n-1}), (W_1, \dots, W_{n-1}), \mathcal{L}, A) & i = n \end{cases} \\ &s_i^{\mathcal{H}}((g, (U_1, \dots, U_n), (W_1, W_2, \dots, W_n), \mathcal{L}, A)) \\ &\quad = \quad (g, (U_1, \dots U_i, \varnothing, U_{i+1}, \dots, U_n), (W_1, \dots W_i, \varnothing, W_{i+1}, \dots, W_n), \mathcal{L}, A) \end{aligned}$$

**Lemma 4.4.** $\mathcal{H}$ is indeed a simplicial set.

*Proof.* Let $\pi_n : \mathcal{H}_n \to N(\mathcal{M})_n$ be the projection onto the interfaces given by $\pi_n((g, r, o, \mathcal{L}, A)) = ((r_i, o_i))_{i \in [n]}$.

*Claim:* the following diagrams commute.

$$\begin{tikzcd}
\mathcal{H}_n \arrow[r, "d_i^{\mathcal{H}}"] \arrow[d, "\pi_n"] & \mathcal{H}_{n-1} \arrow[d, "\pi_{n-1}"] \\
N(\mathcal{M})_n \arrow[r, "d_i^{\mathcal{M}}"] & N(\mathcal{M})_{n-1}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{H}_n \arrow[r, "s_j^{\mathcal{H}}"] \arrow[d, "\pi_n"] & \mathcal{H}_{n+1} \arrow[d, "\pi_{n+1}"] \\
N(\mathcal{M})_n \arrow[r, "s_j^{\mathcal{M}}"] & N(\mathcal{M})_{n+1}
\end{tikzcd}$$

**Face maps**
$i = 0$:

$$\begin{aligned}
&\pi_{n-1}(d_0^{\mathcal{H}}((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A))) \\
&= \pi_{n-1}((g,(U_2,\dots,U_n),(W_2,\dots,W_n),\mathcal{L},A)) \\
&= ((U_2,W_2),\dots,(U_n,W_n)) \\
&= d_0^{\mathcal{M}}((U_1,W_1),\dots,(U_n,W_n)) \\
&= d_0^{\mathcal{M}}(\pi_n((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A)))
\end{aligned}$$

$0 < i < n$:

$$\begin{aligned}
&\pi_{n-1}(d_i^{\mathcal{H}}((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A))) \\
&= \pi_{n-1}((g,(U_1,\dots,U_i\cup U_{i+1},\dots,U_n),(W_1,\dots,W_i\cup W_{i+1},\dots,W_n),\mathcal{L},A)) \\
&= ((U_1,W_1),\dots,(U_i\cup U_{i+1},W_i\cup W_{i+1}),\dots,(U_n,W_n)) \\
&= d_i^{\mathcal{M}}((U_1,W_1),\dots,(U_n,W_n)) \\
&= d_i^{\mathcal{M}}(\pi_n((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A)))
\end{aligned}$$

$i = n$:

$$\begin{aligned}
&\pi_{n-1}(d_n^{\mathcal{H}}((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A))) \\
&= \pi_{n-1}((g,(U_1,\dots,U_{n-1}),(W_1,\dots,W_{n-1}),\mathcal{L},A)) \\
&= ((U_1,W_1),\dots,(U_{n-1},W_{n-1})) \\
&= d_n^{\mathcal{M}}((U_1,W_1),\dots,(U_n,W_n)) \\
&= d_n^{\mathcal{M}}(\pi_n((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A)))
\end{aligned}$$

**Degeneracy maps**

$$\begin{aligned}
&\pi_{n+1}(s_i^{\mathcal{H}}((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A))) \\
&= \pi_{n+1}((g,(U_1,\dots,U_i,\varnothing,U_{i+1},\dots,U_n),(W_1,\dots,W_i,\varnothing,W_{i+1},\dots,W_n),\mathcal{L},A)) \\
&= ((U_1,W_1),\dots,(U_i,W_i),(\varnothing,\varnothing),(U_{i+1},W_{i+1}),\dots,(U_n,W_n)) \\
&= s_i^{\mathcal{M}}((U_1,W_1),\dots,(U_n,W_n)) \\
&= s_i^{\mathcal{M}}(\pi_n((g,(U_1,\dots,U_n),(W_1,\dots,W_n),\mathcal{L},A)))
\end{aligned}$$

Thus $\pi_n$ is a simplicial morphism. Therefore since $N(\mathcal{M})$ is a simplicial set by definition the simplicial identities also hold for $d^{\mathcal{H}}$ and $s^{\mathcal{H}}$. Therefore $\mathcal{H}$ is a simplicial set. $\square$

Note that for a given ranked hypergraph $G$ its underlying DAH $g$ is invariant under $d$ and $s$. This leads to the following result.

**Corollary 4.5.** $\mathcal{H}^1 := HG_1 \cap \mathcal{H}$ is a subobject of $\mathcal{H}$. That is, $\mathcal{H}^1$ is a simplicial set with as face and degeneracy maps $d^{\mathcal{H}}$, respectively $s^{\mathcal{H}}$.

*Proof.* $d^{\mathcal{H}}$, $s^{\mathcal{H}}$ already satisfy the simplicial identities. All that remains to be proven is that $\mathcal{H}^1$ is closed under them. Let $G \in \mathcal{H}^1_n$, then $F := d_i^{\mathcal{H}}(G)$ has $g^F = g^G$ by definition of $d^{\mathcal{H}}$. Therefore $D(F) = D(G) = 1$ and hence $F \in \mathcal{H}^1_{n-1}$. Idem for $s^{\mathcal{H}}$. Thus $\mathcal{H}^1$ is a subobject of $\mathcal{H}$ $\square$

# 5 Towards coalgebraic behaviour

For our setup a coalgebra with well-defined behaviour is not yet found. Therefore in this section we pose a potential coalgebra and a subsequent behaviour function, which has a few problems. Besides this we propose some solutions, which might be worth further investigation.

## 5.1 The idea

To add behaviour to the hypergraph simplicial set we define pointed F-coalgebras by the endofunctor

**Definition 5.1.**

$$F : sSet \to sSet, \quad FX = \Omega \times \mathcal{P}_\omega(X \sqcup \uparrow X)^{HG_1}$$

Through this definition we find out what a coalgebra does on our set. If we follow the repeated iteration of this coalgebra

$$X \xrightarrow{\alpha} FX \xrightarrow{F\alpha} FFX \xrightarrow{FF\alpha} \dots$$

We get by definition of $F(f)$ that $\alpha$ gets recursively applied to the transitioned-to elements. This will look like

$$\begin{aligned} \alpha(x_0) &= (0, \{g_{01} \mapsto x_{01}, g_{02} \mapsto x_{02}, \dots\}) \\ \implies \alpha \circ \alpha(x_0) &= (0, \{(g_{01} \mapsto (0, \{g_{011} \mapsto x_{011}, g_{012} \mapsto x_{012}\})), \\ &\qquad (g_{02} \mapsto (0, \{g_{021} \mapsto x_{021}, g_{022} \mapsto x_{022}\})), \dots\}) \end{aligned}$$

Which is a tree structure with $x_0$ as root, $x_{01}, x_{02}, \dots$ as children etcetera. The transitions are through the hypergraphs. In figure 5 a tree is visualized for some coalgebra. Non-accepting states are red, and accepting states are indicated by a green node. Here $x_{012}$ is an accepting state.

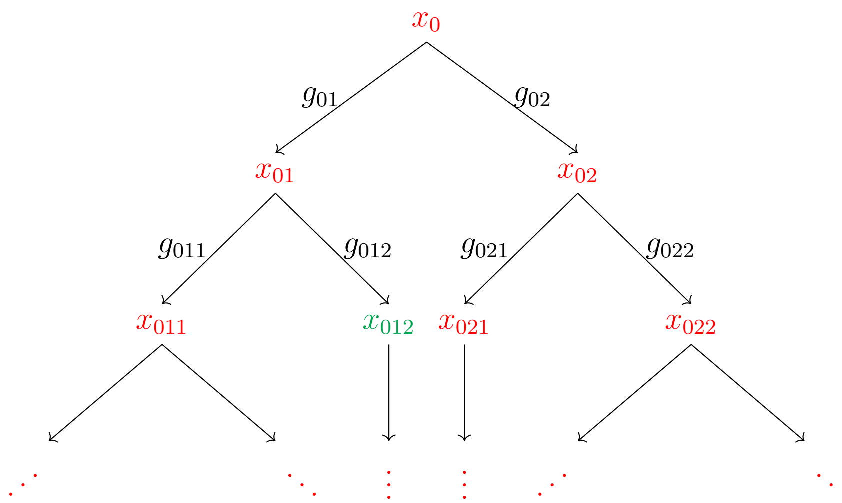


Figure 5: Tree from iteration of some $F$-coalgebra

**Definition 5.2.** Let $(A, \vee, \perp)$ be a join-semilattice. We define the JSL morphism $\bigvee : \mathcal{P}_\omega(A) \to A$ as the map making the following commute:

$$\begin{tikzcd} A \arrow[r, "\mathrm{id}_A"] \arrow[d, "\eta"] & A \\ \mathcal{P}_\omega(A) \arrow[ur, "\bigvee"'] & \end{tikzcd}$$

by lemma 2.5 $\mathcal{P}_\omega(A)$ is the free JSL, hence $\bigvee$ exists and is unique.

**Definition 5.3.** Let $ev : A \times [A, B] \to B$, $(a, f) \mapsto f(a)$ be the evaluation morphism. We extend this to evaluate finite powersets by taking

$$\mathcal{P}_\omega ev : \mathcal{P}_\omega(A) \times [A, B] \to \mathcal{P}_\omega(B)$$
$$\mathcal{P}_\omega ev(\{a_1, \dots, a_n\}, f) = \{f(a_1), \dots, f(a_n)\}$$

To find out if a ranked hypergraph is part of a coalgebras language we split it up into parts of depth 1 using theorem 3.12. Which are then used left to right to transition between states. If any of the final states is accepting, the graph is part of the language.

To formalise this we define a behaviour function.

**Definition 5.4.** Let $(X, c, x_0)$ be a pointed F-coalgebra of 5.1. Define the so called *behaviour function* as

$$\bar{c} : \mathcal{H} \to [X, \Omega]$$
$$\bar{c} = \overline{(\bar{c}_0, \bar{c}_1)}$$

Where $\overline{c}$ is the inductive closure of $\overline{c}_0, \overline{c}_1$:

$$\begin{array}{ll} \overline{c}_0 : \mathbb{1} \to [X,\Omega] & \overline{c}_0 = \lambda(\pi_1 \circ c) \\ \overline{c}_1 : HG_1 \times [X,\Omega] \to [X,\Omega] & \overline{c}_1 = \lambda_2 K \end{array}$$

Where $K$ is a morphism defined by the following commuting diagram

$$\begin{tikzcd} HG_1 \times X \times [X,\Omega] \arrow[r, "K"] \arrow[d, "\widetilde{\pi_2 \circ c} \times \mathrm{id}"'] & \Omega \\ \mathcal{P}_\omega(X \sqcup \uparrow X) \times [X,\Omega] \arrow[r, "\mathcal{P}_\omega ev"] & \mathcal{P}_\omega(\Omega) \arrow[u, "\bigvee"'] \end{tikzcd}$$

## 5.2 Current problems and possible solutions

The problems come in when looking at the bottom arrow of diagram 5.4. Here the carrier of the finite powerset in $\mathcal{P}_\omega(X \sqcup \uparrow X)$ and the domain of the simplicial morphism in $[X,\Omega]$ aren't equal. Thus the use of the evaluation map $\mathcal{P}_\omega ev$ isn't fully defined.

But intuitively it works. A simplicial map is a natural transformation, a family of morphisms such that there is one morphism for each $[n] \in \Delta$. So a potential solution is pointwise, for an $n$-cell in $X$ we transition to a finite set of $n$- and $(n+1)$-cells in $X \sqcup \uparrow X$. For any simplicial morphism $f : X \to S$ we have a function $f_n : X_n \to S_n$ for all $n$. Therefore we could evaluate all transitioned-to elements using a simplicial morphism with $X$ as domain.

A second problem is the disconnect between the generator $HG_1$ and the simplicial set $\mathcal{H}$. At this point in the research we have only found a deconstruction for all hypergraphs into $HG_1$. But this isn't the state space $\mathcal{H}_1$ we wanted to use. A possible solution would be something like $\mathcal{H}^1$, as defined in Corollary 4.5. Or, if there is need for a more drastic change in the simplicial set, consider the following:

Describe a relation of ranked hypergraphs that show they have the same behaviour when looked at as dependency graphs, i.e. the described dependencies and action executions are the same. This is visible in figure 6, in both graphs actions $a$ and $b$ need to be taken to 'fulfill' the actions of the graph, except their graphs structures aren't equal. Up to this relation, there even exists a right identity, which could lead to a monoidal structure. This could in turn be used to define a simplicial set by the nerve construction which would partition the entire set of ranked hypergraphs by their depth.

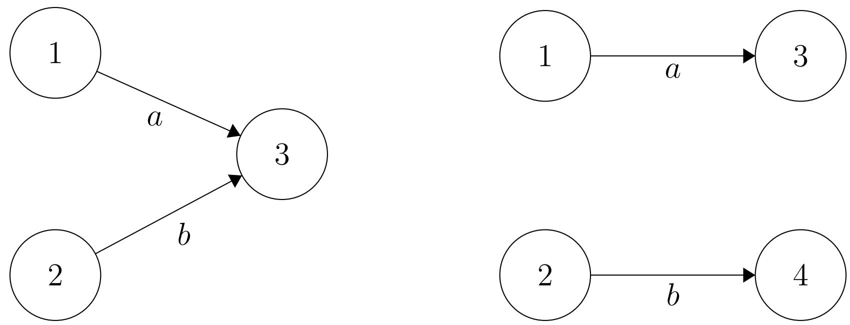


Figure 6: Dependency graphs of equivalent behaviour ($a$ parallel with $b$)

## 6 Conclusions and Further Research

Ranked hypergraphs seem to be a very strong and intuitive model for concurrent behaviour. Their interfaces allow for more complex dependency handling than standard dependency graphs. Associative composition with a left unit are already very strong properties, which combine nicely with quite a simple generator. Moreover the construction in the proof of Theorem 3.12 gives a very computable construction of any graph into a sequence of generating depth-1 graphs.

More work needs to be done primarily on the semantics side as there isn't yet a working behaviour map. To find this there will need to be changes made to at least the coalgebra, if not the entire simplicial set, as is detailed in section 5.2.

## References


[ABC+24] Amrane, Amazigh ; Bazille, Hugo ; Clement, Emily ; Fahrenberg, Uli ; Ziemiański, Krzysztof: Presenting Interval Pomsets with Interfaces. In: *21st International Conference on Relational and Algebraic Methods in Computer Science RAMICS 2024*. Prague, Czech Republic : Springer Nature Switzerland, August 2024

[AHS09] Adamek, Jiri ; Herrlich, Horst ; Strecker, George E.: *Abstract and concrete categories: The joy of cats*. Dover Publications, 2009

[FJSZ24] Fahrenberg, Uli ; Johansen, Christian ; Struth, Georg ; Ziemiański, Krzysztof: Kleene Theorem for Higher-Dimensional Automata. In: *Logical Methods in Computer Science* Volume 20, Issue 4 (2024), Dezember. http://dx.doi.org/10.46298/lmcs-20(4:22)2024. – DOI 10.46298/lmcs–20(4:22)2024. – ISSN 1860–5974

[Gla91] Glabbeek, Rob J.: *Bisimulations for higher dimensional automata*. Email message, Juni 1991. – http://theory.stanford.edu/~rvg/hda

[Joh02] Johnstone, Peter T.: *Sketches of an elephant a topos theory compedium.*. Bd. 1. Clarendon Press, 2002

[Lan98] Lane, Saunders M.: *Graduate Texts in Mathematics*. Bd. 5: *Categories for the Working Mathematician*. 2. New York, NY : Springer Science+Business Media, 1998. – XII + 318 S. http://dx.doi.org/10.1007/978-1-4757-4721-8. http://dx.doi.org/10.1007/978-1-4757-4721-8. – ISBN 978–0–387–98403–2

[Mar11] Martin, John C.: *Introduction to languages and the theory of computation*. McGraw-Hill, 2011

[Pra91] Pratt, Vaughan R.: Modeling concurrency with geometry. In: *POPL*. New York City : ACM Press, 1991, S. 311–322