A DATASTRUCTURE FOR HIGHER SESQUICATEGORIES

CHRISTOPH DORN

ABSTRACT. We define a higher-dimensional analogue of the type of lists in a inductive-recursive fashion. Lists elements will live in a "signature". This signature can be understood as the collection of generating morphisms of a higher sesquicategories (that is, a higher category without the exchange law and its higher dimensional analogues). We further discuss a possible extension of higher lists to "higher lists with duals".

CONTENTS

0. INTRODUCTION AND OVERVIEW

One of the most basic inductive types is the type of lists $\mathsf{list}(C)$ with list elements of type C. It can be defined with two constructors

$$l : \mathsf{list}(C) := \quad | \quad \mathrm{id} \qquad\qquad (0.1)$$
$$| \quad \mathbf{cons}(l, p) \ , \ p : C$$

where id means the empty list, and $\mathbf{cons}(l, p)$ glues an element p of type C to the already existing list l. This definition for instance entails the natural numbers $\mathbb{N} \equiv \mathsf{list}(\{ * \})$ as lists of only one element $*$, or strings $\Gamma^* \equiv \mathsf{list}(\Gamma)$ as lists of elements of some alphabet Γ. A main theme of this work will be the task to find a reasonable and simple generalisation of this inductive concept in higher dimensions.

While the approach presented here is guided by certain geometric principles, in general there is certainly no unique answer to fulfil the task of finding a definition of 'higher lists'. For instance, a well-known but quite different approach to 'higher lists' is that of indexed W-types also known as indexed containers or polynomial functors: More precisely the definition uses the slice construction on cartesian polynomial monads . It was shown in that the opetopic approach to higher categories can be based on this construction. However, the opetopic approach does not support a symmetric treatment of input and output. Indeed, the polynomial functor approach more accurately describes what is often called *higher trees* (which can be regarded as a specific flavor of higher lists).

The present work arose from an attempt to 'symmetrise' the (polynomial functor approach of the) *opetopic framework*. This framework was already implemented by the proof assistant opetopic . The original ambition to look for such a symmetrised version of opetopic was the insight that it would lead to a nice graphical representation for a multitude of logics ranging from classical, non-linear, non-polarized to linear, intuitionistic, polarized with relevance to process calculi − this representation will be the subject of future work. Especially the classical aspects of logic are not

Date: September 5, 2016.

in line with the asymmetric approach of opetopes: From a computational perspective, classical logic treats values and continuations (or co-values) on equal footing (cf. Wadler's dual calculus) while on the intuitionistic side co-values are essentially absent (cf. the λ-calculus). Co-values are also essentially absent from mainstream mathematical thinking, while in theoretical computer science they are more well-known and used e.g. in the CPS translation . However, duality has proven a powerful tool in mathematics propelled by the development of category theory and this possibly means that the strong bias towards values in most mathematical foundations should be re-thought. In fact, as pointed out on the n-Category Cafe there is a close connection between the CPS translation and the Yoneda embedding. It shouldn't come as a too big surprise then that by taking 'co-values and co-variables seriously' we will arrive at a particularly nice formulation of topics ranging from the Yoneda Lemma to Equipments.

In section 1 the formal definition of our approach to 'higher lists', which capture higher dimensional behaviour of lists. In light of (0.2), the role of list (in dimension n) will be played by the type of n-diagrams D_n, while the role of C in $\mathsf{list}(C)$ (in dimension n) will be played by the type of n-cells C_n. In fact the definition of D_n will be of the form

$$d : D_n := \quad | \quad \mathrm{id} \qquad | \quad d \underset{c}{\triangleright} p \ , \ p : C, \ P(d, c, p) \tag{0.2}$$

where d is an existing diagram and p gets glued to d as an additional list element: The central difference to (0.2) is that we are not agnostic to *where* p is glued to d anymore, and this 'position of gluing' is described by coordinates c. The condition $P(d, c, p)$ will then express that corresponding boundaries of d and p coincide so that the gluing can in fact take place.

The style in which the definition of D_n, C_n is presented in section 1 is inductive in n but also *inductive-recursive* at each dimension: This means D_n, C_n are defined simultaneously with functions on them, which are required to express $P(d, c, p)$ in simple terms. It should also be noted that both D_n, C_n are further dependent on an n-signature of type Sig_n, which will be defined as well in the inductive process. The largest part of section 1 will be concerned with proving the inductive hypotheses 1.10, proving correctness of the given definitions, and introducing conventions and tools to work with them.

In section 2 we then go on to consider the following extensions of higher lists to 'higher lists with duals', using the following three steps.

(i) *Compositionality.* Compositionality means to extend 'higher lists' to 'higher lists of lists', i.e. lists can have lists as list elements, or conversely, lists compose to yield list elements. Importantly, there will be a higher-dimensional witnesses of such compositions to keep track of this process internally.
(ii) *Duality.* Duality means that we extend our consideration of list elements to that of *co-elements*: Intuitively, while elements are resources or 'values', co-elements can be thought of as deficits or 'continuations' (or 'co-values') indicating that a value is stilled owed or yet to be provided. Operations to both create and compensate deficits will be provided and will be called *shifts*. The classical analogue is that of dual spaces, e.g. $x^* \in X^* := X \to \mathbb{R}$ 'compensates' $x \in X$ by $x^*(x) = 1$. Importantly, deficits themselves qualify as elements and thus we can form 'deficits of deficits' (e.g. X^{**}) and so on: That is $(-)^*$ is not involutive, but we will see $X \cong X^{**}$ naturally.
(iii) *Unitarity.* Unitarity combines compositionality and duality, in that it qualifies resources as 'unitary' if and only if their corresponding shifts are witnesses of composition which in turn is the case iff they are unitary (in this sense, unitarity is a coinductive definition).

1. Higher lists

The definition of n-cells C_n and n-diagrams D_n is both inductive in the dimension n and of inductive-recursive flavour at each dimension: This means, C_n and D_n as well as certain functions on these types will be defined *simultaneously*. Further, we will later introduce paths in the path type Id_{C_n}, making the definitions below 'higher inductive-recursive' (on the other hand, C_n, D_n will have decidable equality making them *sets* and not higher types by Hedberg's Theorem: Thus, the reader should understand Id_{C_n} as an equivalence relation on C_n). Bundling all definitions together, the data structure described in the following could be formalised as an higher indexed inductive-recursive definition. However, we will be agnostic to which specific type theory could be used for this formalisation.

In later section 2 we will extend the 'core' definitions presented here with additional constructors and equalities, to which (we claim) the core definitional process can be straight-forwardly extended. In both the core and the extended versions, the following entities will be defined in the n-th step of the definitional process

$$\begin{aligned} &\mathrm{Sig}_n : \mathrm{Type} && (n\text{-signatures}) \\ &\pi_1^{(n)} : \mathrm{Sig}_n \to \mathrm{Sig}_{n-1} && (\text{signature projection}) \end{aligned} \tag{1.1}$$

and assuming a signature $\sigma_n : \mathrm{Sig}_n$ as a parameter we will further define

$$\begin{aligned} C_n(\sigma_n) &: \mathrm{Type} && (n\text{-cells}) \\ \partial_i^{(n)}(\sigma_n) &: C_n(\sigma_n) \to D_{n-1}(\sigma_n) && (\text{cell boundaries, } i \in \{\, 0, 1 \,\}) \end{aligned} \tag{1.2}$$

$$\begin{aligned} D_n(\sigma_n) &: \mathrm{Type} && (n\text{-diagrams}) \\ |-|^{(n)}(\sigma_n) &: D_n(\sigma_n) \to \mathbb{N} && (\text{diagram size}) \\ \partial_i^{(n)}(\sigma_n) &: D_n(\sigma_n) \to (D_{n-1}(\pi_1\sigma_n))_\perp && (\text{diagram boundaries}) \\ p_i^{(n)}(\sigma_n) &: D_n(\sigma_n) \to (C_n(\sigma_n))_\perp && (\text{diagram processeses}) \\ c_i^{(n)}(\sigma_n) &: D_n(\sigma_n) \to \left(\mathbb{N}^{2n}\right)_\perp && (\text{diagram coordinates}) \end{aligned} \tag{1.3}$$

$$\begin{aligned} (-.-)^{(n)}(\sigma_n) &: D_n(\sigma_n) \times D_n(\sigma_n) \to (D_n(\sigma_n))_\perp && (\text{concatenation}) \\ -|_c^{(n)}(\sigma_n) &: D_n(\sigma_n) \to (D_n(\sigma_n))_\perp && (\text{restriction to } c : \mathbb{N}^{2n}) \\ (- \underset{c}{\blacktriangleright} -)^{(n)}(\sigma_n) &: D_{n-1}(\pi_1\sigma_n) \times D_n(\sigma_n) \to (D_n(\sigma_n))_\perp && (\text{whiskering at } c : \mathbb{N}^{2n}) \end{aligned} \tag{1.4}$$

A few important remarks:

Remark 1.5. (i) *Partiality* is an issue for the operations in (1.3) and (1.4): Recall that the notation $f : A \to B_\perp$ means a function from A into the flat domain obtained from B by adding a bottom element $B_\perp = B + \{\, \perp \,\}$, with the extra structure of partial order: $\forall b : B. \perp < b$. Such an f describes a partial function on A, i.e. one that has either a *defined* value $f(a) : B$ or is *undefined* $f(a) = \perp$, for $a \in A$.

Since the present context is type theory and not domain theory, there are no underlying orders on types and no restriction on monotonicity of functions. However, for the case of flat domains only we can assert monotonicity easily by the following convention: All function definitions of the above form

$$f : A \to B_\perp$$

will be tacitly extended to

$$f : A_\perp \to B_\perp$$

by setting $f(\perp) = \perp$. For functions f with multiple parameters we use the *smash product* $(A_\perp \wedge A'_\perp)$ of pointed domains, which is the product $(A_\perp \times A'_\perp)$ up to the identification $\perp = (a, \perp) = (\perp, a') \in (A_\perp \wedge A'_\perp) \cong (A \times A')_\perp$. For example, in the case of the restriction function from (1.4) this convention implies

$$\bot|_c^{(n)}(\sigma_n) = d|_\bot^{(n)}(\sigma_n) = \bot : (D_n(\sigma_n))_\bot$$

It is important to note that we regard the coordinate $c : \mathbb{N}^{2n}$ as another parameter to the restriction function $-|_c$.

We emphasise that the technicality of using 'partial functions' does not carry real importance to the bigger picture of our discussion and could probably be dealt with otherwise.

(ii) The dependence on the parameter σ_n above could of course also be written out as a dependent type

$$\begin{aligned}
n\text{-Sig} &: \text{Type} && n\text{-signatures} \\
C_n &: \prod_{\sigma_n : n\text{-Sig}} \text{Type} && (n\text{-cells}) \\
\partial_0^{(n)}, \partial_1^{(n)} &: \prod_{\sigma_n : n\text{-Sig}} (C_n(\sigma_n) \to D_{n-1}(\sigma_n)) && (n\text{-cell boundaries}) \\
\\
D_n &: \prod_{\sigma_n : n\text{-Sig}} \text{Type} && (n\text{-diagrams}) \\
|-|^{(n)} &: \prod_{\sigma_n : n\text{-Sig}} (D_n(\sigma_n) \to \mathbb{N}) && (\text{diagram size}) \\
&\dots
\end{aligned}$$

However, for clarity in notation, we will usually assume the dependence on an n-signature implicitly, leading us to drop $\sigma_n : \mathrm{Sig}_n$ from our notation as follows:

$$\begin{aligned}
C_n &: \text{Type} && (n\text{-cells}) \\
\partial_0^{(n)}, \partial_1^{(n)} &: C_n \to D_{n-1} && (n\text{-cell boundaries}) \\
\\
D_n &: \text{Type} && (n\text{-diagrams}) \\
|-|^{(n)} &: D_n \to \mathbb{N} && (\text{diagram size}) \\
&\dots
\end{aligned}$$

(iii) To further lighten notation, we will usually keep the dimension n at which functions act implicit (as these can be inferred from their arguments). Thus we can simplify the above to

$$\begin{aligned}
C_n &: \text{Type} && (n\text{-cells}) \\
\partial_0, \partial_1 &: C_n \to D_{n-1} && (n\text{-cell boundaries}) \\
\\
D_n &: \text{Type} && (n\text{-diagrams}) \\
|-| &: D_n \to \mathbb{N} && (\text{diagram size}) \\
&\dots
\end{aligned}$$

(iv) We indicated in the beginning of this section that the above definitions can be bundled together into an indexed inductive-recursive definition: This means, for instance the definition of D_n will have dependency on parameters from D_{n-1}, C_n as well as dependency on the functions of $\partial_i^{(n)}$, $|-|^{(n)}$, $-|_c^{(n-1)}$. Among these dependencies, C_n, $\partial_i^{(n)}$, $|-|^{(n)}$ will have to be defined simultaneously with D_n as they lie on the same dimension n as D_n. Also note that the definition of n-signatures Sig_n will in fact depend on the definitions of $D_{n-1}, \partial_i^{(n-1)}, |-|^{(n-1)}$ making Sig_n part of the inductive process in n, but no mutual induction with other definitions at dimension n will be required.

(v) C_n, D_n will have decidable equality by virtue of their construction, and thus they are sets by Hedberg's Theorem. We will interchangeably use ':' and '$\in$' for sets, e.g. write both $p : C_n$ and $p \in C_n$. ♢

1.1. **0-cells and 0-diagrams.**

We give the above definitions (1.1), (1.2), (1.3) and (1.4) for the induction start $n = 0$. The definitions will become clearer and more detailed explanation will be provided once we do the inductive step to k-cells and k-diagrams.

(0) First, we note that some of the definitions for the case $n = 0$ require reference to '(-1)-dimensional diagrams' D_{-1} and '(-1)-signatures' Sig_{-1}. We thus set

$$\begin{aligned} D_{-1} &= \{ \epsilon \} \\ \mathrm{Sig}_{-1} &= \{ \epsilon \} \end{aligned}$$

Well will call ϵ the empty cell or empty diagram. We also define size, $|-|$, boundary ∂_i and restriction functions $-|_0$ of (-1)-diagrams

$$\begin{aligned} |-| \ &: D_{-1} \to \mathbb{N} \\ &\epsilon \mapsto |\epsilon| = 0 \\ \partial_i \ &: D_{-1} \to \{ \perp \} \\ &\epsilon \mapsto \partial_i \epsilon = \perp \\ -|_0 \ &: D_{-1} \to D_{-1} \\ &\epsilon \mapsto \epsilon|_0 = \epsilon \end{aligned}$$

(i) **Signatures Sig_n (1.1):** For $n = 0$ the defintions for (1.1) are as follows. A 0-signature σ_0 is essentially a set A with decidable equality, denoted A : dec-Set. However, we will also keep track of the (-1)-signature (there is only the empty one, namely ϵ) and the signature projection to dimension -1 is the projection to this information. Explicitly, we set

$$\begin{aligned} \mathrm{Sig}_0 &:= \mathrm{Sig}_{-1} + \text{dec-Set} \\ \pi_1(\sigma_0 : \mathrm{Sig}_0) : \mathrm{Sig}_{-1} &:= \epsilon \end{aligned}$$

For a 0-signature $\sigma_0 = (\epsilon, A)$ we will often refer to $A = \pi_2 \sigma_0$ by G_0: the set of '0-generators'.

(ii) **Cells C_0 (1.2):** For $n = 0$ the defintions for (1.2) are as follows. For now, the only way to construct a cell is from a generator:

$$p \in C_0 := \quad | \quad \mathrm{cell}(g) \ , \ g \in G_0$$

The boundaries of these cells are given by maps $\partial_0, \partial_1 : C_0 \to D_{-1}$ such that

$$\partial_0 \partial_1 (p : C_0) : D_{-1} := \quad \epsilon$$

(ii)b **Cell identities Id_{C_0}:** The identity type Id_{C_0} is the trivial one, meaning that it does not contain paths/equations apart from witnesses of reflexivity $\mathrm{refl}_p : p = p$.

(iii) **Diagrams D_0 (1.3):** For $n = 0$ the definitions are as follows: 0-Diagrams are either identities (written $\mathrm{id}(b)$) of lower dimensional diagrams, or diagrams glued with a 0-cell (written p $(d \triangleright p)$):

$$\begin{aligned} d \in D_0 := \quad & | \quad \mathrm{id}(b) \quad , \ b \in D_{-1} \\ & | \quad d \underset{0}{\triangleright} p \ , \ p \in C_0 \\ & \qquad \textbf{if } \left[\left(\partial_{|d|} d \right)\big|_0 = \partial_0 p \right] \end{aligned}$$

Remark 1.6. The following points should be noted

(a) $[A]$ denotes the support of a type A, i.e. the proposition isInhabited(A) : Prop.

(b) We will sometimes also write $d \underset{0}{\triangleright} p$ as $\mathbf{cons}(d, p, 0, w)$ to firstly, emphasise that it has to be read as a constructor, and secondly, to remind ourselves that the constructor implicitly also depends on a witness

$$w : \left[\left(\partial_{|d|} d\right)\big|_0 = \partial_0 p \right] : \text{Prop}$$

as a parameter.

(c) It is 'legitimate' in the following sense to drop w from our notation $d \underset{0}{\triangleright} p$ above: Since a type of the form $[a = b]$ is a proposition, we know that any two witnesses w, w' can be (internally) equated and so $\mathbf{cons}(d, p, 0, w), \mathbf{cons}(d, p, 0, w')$ will be equal, too. We will thus regard $\left[\left(\partial_{|d|} d\right)\big|_0 = \partial_0 p \right]$ as a 'side condition' for the constructor $d \underset{0}{\triangleright} p$ to be applicable.

(d) For general dimension n, the number below '$\triangleright$' will denote a coordinate vector in $\mathbb{N}^{2n}$, but in the 0-dimensional case these coordinates must always be the origin $0 \in \mathbb{N}^0$ − In the case of a coordinate being the origin we usually do not annotate it and keep the additional vector $\vec{0}$ tacit writing $\triangleright$ instead of $\underset{\vec{0}}{\triangleright}$.

(e) Further note that the side condition will in fact turn out to be always true in the 0-dimensional case. To see this we still need to define the functions that were used in it, namely ∂_i and $|{-}|$. ♢

Before we define these functions, we recall from the introduction that most definitions of functions on inductive datatypes T will be given by some kind of pattern matching with the following notation: To define a function $f : T \to T'$ we write

$$f(t : T) : T' := \quad | \quad t = \text{constructor}(a_1, ...a_i) \ \mapsto \text{value}(a_1, ..., a_i) \\ | \ \dots$$

to mean $f(t) := \text{value}(a_1, ..., a_i)$ if t matches constructor$(a_1, ...a_i)$. If $T' = (T'')_\perp$ we implicitly add the case $t = \perp \ \mapsto \ \perp$ as was described in Remark 1.5.

Using this notation, the size, process and coordinate maps are given by

$$|d : D_0| : \mathbb{N} := \quad | \quad d = \text{id}(b) \ \mapsto 0 \\ | \quad d = d' \triangleright p \ \mapsto |d'| + 1$$

$$p_i(d : D_0) : (C_0)_\perp := \quad | \quad d = \text{id}(b) \ \mapsto \perp \\ | \quad d = d' \triangleright p \ \mapsto \begin{cases} p & i = |d'| + 1 \\ p_i(d) & \text{otherwise} \end{cases}$$

$$c_i(d : D_0) : (\mathbb{N}^0)_\perp := \quad | \quad d = \text{id}(b) \ \mapsto \perp \\ | \quad d = d' \triangleright p \ \mapsto \begin{cases} 0 & i = |d| + 1 \\ c_i(d') & \text{otherwise} \end{cases}$$

The boundary maps for diagrams are given by

$$\partial_i(d : D_0) : (D_{-1})_\perp := \begin{cases} \epsilon & 0 \le i \le |d| \\ \perp & \text{otherwise} \end{cases}$$

Remark 1.7. Recall from Remark 1.5 that the notation $f : A \to B_\perp$ means a function from A into the flat domain of B, which effectively describes a partial function, i.e. one that has either a value $f(a) \in B$ or is *undefined* $f(a) = \perp$ for $a \in A$. Let $\text{supp}(f) \subset A$ be the subset where f is defined, which we call the support of f. By induction on the strucutre of $d \in D_0$ we see that

$$\mathrm{supp}(\lambda(i:\mathbb{N}).p_i d) = \mathrm{supp}(\lambda(i:\mathbb{N}).c_i d) = \{\, i \mid 1 \le i \le |d| \,\}$$
$$\mathrm{supp}(\lambda(i:\mathbb{N}).\partial_i d) = \{\, i \mid 0 \le i \le |d| \,\}$$

$\diamondsuit$

(iii)b **Diagram identities** Id_{D_0}**:** As for C_0 there are no non-trivial equality paths on D_0. However it should be pointed out that equality has a description in terms of the functions ∂_i, p_i, c_i as follows

$$d_1 = d_2 \ , \ d_i \in D_0$$
$$\iff \forall i.(\partial_i d_1, p_i d_1, c_i d_1) = (\partial_i d_2, p_i d_2, c_i d_2)$$

This could be proven by induction on the structure of $d_i \in D_0$. However, in the present case of dimension 0, in fact neither ∂_i nor c_i play a role for equality, as both are constant on their support. Instead, we can identify 0-diagrams essentially with lists of programs writing

$$d \equiv (p_1 d, p_2 d ..., p_{|d|} d)$$

for

$$d \equiv (...((\mathrm{id}(\epsilon) \underset{0}{\triangleright} p_1 d) \underset{0}{\triangleright} p_2 d) ... \underset{0}{\triangleright} p_{|d|} d) \ .$$

In the higher dimensional case, it will become useful to keep track of boundaries and coordinates and for this purpose we already introduce the following notation

$$d \equiv \ \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \ \cdots \ \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d$$

Example 1.8. In the present case of dimension 0 this means we write

$$d \equiv \ \epsilon \xrightarrow[0]{p_1 d} \epsilon \xrightarrow[0]{p_2 d} \epsilon \ \cdots \ \epsilon \xrightarrow[0]{p_{|d|} d} \epsilon$$

for

$$d \equiv (...((\mathrm{id}(\epsilon) \underset{0}{\triangleright} p_1 d) \underset{0}{\triangleright} p_2 d) ... \underset{0}{\triangleright} p_{|d|} d) \ .$$

Remark 1.9. In the upcoming sections, the equality type Id_{C_n} will be augmented by new equivalences. It is important to note that adding new equalities on types is only admissible and consistent if all previously defined functions (e.g. ∂_i, p_i, c_i) on these types *preserve the new equalities*, i.e. if they stay 'well-defined'. $\diamondsuit$

(iv) **Diagram operations** (1.4)**:** For $n = 0$, we define
- a function $(-.-)$, called concatenation, which concatenates two 0-diagrams along a common boundary,
- a function $-|_c$, $c \in \mathbb{N}^2$, called restriction, that restricts a 0-diagram to a sub-diagram described by the coordinates c,
- and a function $(- \underset{0}{\blacktriangleright} -)$, called whiskering, which whiskers a diagram $d \in D_0$ by the empty boundary $b \in D_{-1}$ at coordinates 0.

For the first we set:

$$\begin{aligned} -.- \ : \ & D_0 \times D_0 \to (D_0)_\perp \\ & (d, d') \equiv ((p_1 d, ..., p_{|d|} d), (p_1 d' ..., p_{|d'|} d')) \\ & \quad \mapsto (p_1 d, ..., p_{|d|} d, p_1 d' ..., p_{|d'|} d') \end{aligned}$$

i.e. in the 0-dimensional case $-.-$ is in fact the usual concatenation of lists, and the common boundary is always the empty ϵ.

Restriction is defined by

$$-|_{(l,r)} : \ D_0 \to (D_0)_\perp$$
$$(d) \equiv (p_1 d, ..., p_{|d|} d) \mapsto \begin{cases} (p_{l+1}, ..., p_{|d|-r}) & l + r \leq |d| \\ \perp & \text{otherwise} \end{cases}$$

Lastly, we consider the operation of 'whiskering diagrams' at coordinates $0 \in \mathbb{N}^0$:

$$- \underset{0}{\blacktriangleright} - \ : \ D_{-1} \times D_0 \to (D_0)_\perp$$
$$(b, d) \equiv (\epsilon, (p_1 d, ..., p_{|d|} d))$$
$$\mapsto (p_1 d, ..., p_{|d|} d) \equiv d$$

i.e. in the 0-dimensional case $- \underset{0}{\blacktriangleright} -$ is in fact only gluing a list to the empty boundary, which yields again the list that we started with.

Importantly, these operations satisfy the following Inductive Claim 1.10 for $n = 0$, which summarises the inductive hypothesis that we will be working with. The reader should convince herself that indeed all claims for the case $n = 0$ follow from the above definitions.

Inductive Claim 1.10. *The definitions of the terms stated in (1.1), (1.2), (1.3) and (1.4) satisfy the following*

(i) For $p \in C_n$, we have

$$\partial_0 \partial_0 p = \partial_0 \partial_1 p$$
$$\partial_{|\partial_0 p|} \partial_0 p = \partial_{|\partial_1 p|} \partial_1 p \tag{1.11}$$

called the 'globular conditions'.

(ii) For $d_1, d_2 \in D_n$ we have

$$d_1 = d_2$$
$$\iff \forall i.(\partial_i d_1, p_i d_1, c_i d_1) = (\partial_i d_2, p_i d_2, c_i d_2)$$

and moreover

$$\text{supp}(\lambda(i : \mathbb{N}).p_i d) = \text{supp}(\lambda(i : \mathbb{N}).c_i d) = \{ \ i \mid 1 \leq i \leq |d| \ \}$$
$$\text{supp}(\lambda(i : \mathbb{N}).\partial_i d) = \{ \ i \mid 0 \leq i \leq |d| \ \}$$

Thus we can write

$$d \equiv \ \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \ \cdots \ \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d \ \in D_k$$

in order to refer to a diagram d in D_n.

(iii) Under the assumption of part (ii) the following holds.

(a) (Whiskering) *Given $d = \ \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \ \cdots \ \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_s d} \partial_{|d|} d \ \in D_n$, and $b \in D_{n-1}$, then $b \underset{c}{\blacktriangleright} d$ is defined if and only if $b|_c = \partial_0 d$ and in this case we have:*

$$b \underset{c}{\blacktriangleright} d = \ b \xrightarrow[c + c_1 d]{p_1 d} b_1 \xrightarrow[c + c_2 d]{p_2 d} b_2 \ \cdots \ b_{|d|-1} \xrightarrow[c + c_{|d|} d]{p_{|d|} d} b_{|d|}$$

where $b_i = \partial_i (b \underset{c}{\blacktriangleright} d) \in D_{n-1}$, $i \geq 1$, and $c + c_i$ denotes vector addition (cf. Remark 1.13).

(b) (Concatenation) *For $d, d' \in D_n$, $d.d'$ is defined if and only if $\partial_{|d|} d = \partial_0 d'$ and in this case we have:*

$$d.d' = \ \partial_0 d \xrightarrow[c_1 d]{p_1 d} \cdots \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d \xrightarrow[c_1 d']{p_1 d'} \cdots \xrightarrow[c_{|d'|} d']{p_{|d'|} d'} \partial_{|d'|} d'$$

Note that this makes $(-.-)$ associative and thus we can write e.g. $d.d'.d''$ without brackets.

(c) (Restriction) *For $d, d' \in D_n$ we have*

$$d|_{(c,l,r)} = d' \iff d = d_L.(\partial_l d \underset{c}{\blacktriangleright} d').d_R \ , \ \textit{where } |d_L| = l, |d_R| = r \tag{1.12}$$

(d) (Distributivity) *For $d, d' \in D_n$, $b \in D_{n-1}$ we have*

$$b \underset{c}{\blacktriangleright} (d.d') = \left(b \underset{c}{\blacktriangleright} d\right).\left(b' \underset{c}{\blacktriangleright} d'\right)$$

where $b' = \partial_{|d|}(b \underset{c}{\blacktriangleright} d)$.

(e) (Stable Boundaries) *For $d, d' \in D_n$ with $\partial_0 d = \partial_0 d'$ and $\partial_{|d|} d = \partial_{|d'|} d'$ and $b \in D_{n-1}$ we have*

$$\partial_{|d|}(b \underset{c}{\blacktriangleright} d) = \partial_{|d'|}(b \underset{c}{\blacktriangleright} d')$$

in case either the left or the right hand side is defined.

Note that (Restriction) does not directly define how to compute the restriction function, but describes restriction in terms of the concatenation and whiskering: It expresses that d' is a sub-diagram ('a restriction') of d, if and only if d can be obtained from d' by appropriate gluing and concatenating with the 'missing parts' of d (right hand side).

Remark 1.13. We use vector addition and substraction $+, - : \mathbb{N}^{2n} \times \mathbb{N}^{2n} \to (\mathbb{N})_\perp$ on coordinates: While addition of two natural number vectors is always defined, substraction of two vectors is only defined if it is elementwise *non-negative*. ♢

Lemma 1.14. *The Inductive Claim 1.10 holds for $n = 0$.*

Proof. Some claims have already been discussed explicitly. Since for $n = 0$ we observed that D_n is just the type of lists of 0-cells (and $(-.-)$ is list concatenation, $-|_c$ is restriction to sub-lists and $\epsilon \blacktriangleright -$ is the identity), all remaining claims can be 'read off' from the definitions. □

1.2. k-cells and k-diagrams.

Let $k > 0$. We assume that, for all $n < k$, all terms in (1.1), (1.2), (1.3) and (1.4) have been defined and satisfy the Inductive Claim 1.10.

(i) **k-Signatures** Sig_k (1.1)**:** For a set G_k : dec-Set with decidable equality and equipped with maps $\partial_0, \partial_1 : G_k \to C_{k-1}$ we define the *k-globular condition* $\mathrm{glob}_k(G_k, \partial_0, \partial_1)$ to be satisfaction (i.e. inhabitation) of the following equalities:

$$\mathrm{glob}_k(G_k, \partial_0, \partial_1) := \prod_{g:G_k} \begin{bmatrix} \partial_0\partial_0 g = \partial_0\partial_1 g \\ \wedge\ \partial_{|\partial_0 g|}\partial_0 g = \partial_{|\partial_1 g|}\partial_1 g \end{bmatrix}$$

We then define k-generator sets Gen_k to be the type of such sets

$$\mathrm{Gen}_k := \sum_{G_k:\text{dec-Set}} \ \sum_{\partial_0,\partial_1:G_k \to C_{k-1}} \mathrm{glob}_k(G, \partial_0, \partial_1)$$

Finally a k-signature is given by a $(k-1)$-signature together with a k-generator set, yielding the following type of k-signatures

$$\mathrm{Sig}_k := \mathrm{Sig}_{k-1} + \mathrm{Gen}_k$$

We define the signature projection $\pi_1 : \mathrm{Sig}_k \to \mathrm{Sig}_{k-1}$ to be the actual projection on the first component:

$$\begin{aligned} \pi_1 \ &: \mathrm{Sig}_k \to \mathrm{Sig}_{k-1} \\ &\sigma_k = (\sigma_{k-1}, a : \mathrm{Gen}_k) \quad \mapsto \quad \sigma_{k-1} \end{aligned}$$

Below we will implicitly assume a signature $\sigma_k = (\sigma_k, a)$ (as was already discussed in Remark 1.5). Further we will refer to the set of k-generators $\pi_1 a$ as G_k.

(ii) **k-Cells C_k (1.2):** As in the 0-dimensional case, for now, the only way we will allow cells to be constructed is from generators

$$p \in C_k := \quad | \quad \mathrm{cell}(g)\ ,\ g \in G_k$$

The boundary maps $\partial_0, \partial_1 : C_k \to D_{k-1} \times D_{k-1}$ for k-cells of this form are given by pattern matching

$$\partial_0(p : C_k) : D_{k-1} \times D_{k-1} := \quad | \quad \mathrm{cell}(g) \ \mapsto \partial_0 g$$

and

$$\partial_1(p : C_k) : D_{k-1} \times D_{k-1} := \quad | \quad \mathrm{cell}(g) \ \mapsto \partial_1 g$$

where on the right hand sides ∂_i denotes the boundary function provided by the signature's k-generator set.

Claim 1.15. *All $p \in C_k$ satisfy the globular conditions* (1.11) *(for $n = k$).*

Proof. By definition of ∂_0, ∂_1, this property is inherited from the globular condition on generators $\mathrm{glob}_k(G_k, \partial_0, \partial_1)$. □

(ii)b **k-Cell identities Id_{C_k}:** For now, we do not introduce non-trivial identity paths on C_k.

(iii) **k-Diagrams D_k (1.3):** Next we define k-diagrams: As before diagrams are either identities on lower dimensional boundaries, or they arise from gluing a process p to a diagram d. Unlike the 0-dimensional case however, the coordinates c 'where' to glue the process are now non-trivial vectors in $\mathbb{N}^{2k}$ and implicate a typing condition for gluing: The sub-boundary in $\partial_{|d|} d$ described by c, should agree with the boundary $\partial_0 p$ in order for the gluing to be well-typed. This is captured in the following definition

$$\begin{aligned} d \in D_k := \quad & | \quad \mathrm{id}(b) \quad , \ b \in D_{k-1} \\ & | \quad d \underset{c}{\triangleright} p \ , \ p \in C_k, c \in \mathbb{N}^{2k} \\ & \qquad\qquad \textbf{if } \left[\partial_{|d|} d \big|_c = \partial_0 p \right] \end{aligned}$$

As before we let $d \underset{c}{\triangleright} p$ be equivalently denoted by $\mathbf{cons}\left(d, c, p, w : \left[\partial_{|d|} d\big|_c = \partial_0 p\right]\right)$ in order to keep track of the **if**-condition (cf. Remark 1.6).

Remark 1.16. We inductively define the empty k-diagram $\epsilon^{(k)} \in D_k$ by

$$\epsilon^{(k)} := \mathrm{id}\left(\epsilon^{(k-1)}\right)$$

with $\epsilon^{(-1)} := \epsilon \in D_{-1}$ to start the induction. By convention, we will keep the dimension k implicit, and just write $\epsilon \in D_k$.

The size, process and coordinate functions are defined as for the case $k = 0$:

$$\begin{aligned} |d' : D_k| : \mathbb{N} := \quad & | \quad d' = \mathrm{id}(b) \quad \mapsto 0 \\ & | \quad d' = d \underset{c}{\triangleright} p \ \mapsto |d| + 1 \end{aligned}$$

$$\begin{aligned} p_i(d' : D_k) : (C_k)_\perp := \quad & | \quad d' = \mathrm{id}(b) \quad \mapsto \perp \\ & | \quad d' = d \underset{c}{\triangleright} p \ \mapsto \begin{cases} p & i = |d'| + 1 \\ p_i(d) & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} c_i(d' : D_k) : (\mathbb{N}^{2k})_\perp := \quad & \\ & | \quad d' = \mathrm{id}(b) \quad \mapsto \perp \\ & | \quad d' = d \underset{c}{\triangleright} p \ \mapsto \begin{cases} c & i = |d'| + 1 \\ c_i(d) & \text{otherwise} \end{cases} \end{aligned}$$

For the boundaries we need to refer to the inductive assumption (1.12) which applied to the side condition on the $\underset{c}{\triangleright}$ constructor says that (matching c with $(c', l, r), c' \in \mathbb{N}^{2(n-1)}$):

$$\partial_{|d|}d\big|_{(c',l,r)} = \partial_0 p$$
$$\iff \partial_{|d|}d = \left(\partial_{|d|}d\right)_L . (\partial_l\partial_{|d|}d \underset{c}{\blacktriangleright} \partial_0 p). \left(\partial_{|d|}d\right)_R \ \text{, where } \left|\left(\partial_{|d|}d\right)_L\right| = l, \left|\left(\partial_{|d|}d\right)_R\right| = r$$

With this in mind (and retaining the above notation for $\left(\partial_{|d|}d\right)_L$ and $\left(\partial_{|d|}d\right)_R$) we define

(1.17)

$\partial_i(d' : D_k) : (D_{k-1})_\perp :=$

$$\mid \quad d' = \mathrm{id(b)} \qquad \mapsto \begin{cases} b & i = 0 \\ \perp & \text{otherwise} \end{cases}$$

with $c = (c', l, r), c' \in \mathbb{N}^{2(n-1)}$:

$$\mid \quad d' = \mathbf{cons}\left(d, c, p, w : \left[\partial_{|d|}d\big|_c = \partial_0 p\right]\right) \mapsto \begin{cases} \left(\partial_{|d|}d\right)_L . (\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p). \left(\partial_{|d|}d\right)_R & i = |d'| + 1 \\ \partial_i d & \text{otherwise} \end{cases}$$

We note that the **with** clause indicates that we performed an intermediate pattern matching on c and set c to be of the pattern (c', l, r).

Claim 1.18. *We claim that in the above context*

$$\left(\partial_{|d|}d\right)_L . (\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p) . \left(\partial_{|d|}d\right)_R \tag{1.19}$$

is in fact a defined diagram.

Proof. We have to show definedness of the three operations marked in red. We know that

$$\left(\partial_{|d|}d\right)_L . (\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_0 p). \left(\partial_{|d|}d\right)_R$$

is a well-defined diagram, namely it equals $\partial_{|d|}d$. The globular conditions on C_k say that both $\partial_0\partial_0 p = \partial_0\partial_1 p$ and $\partial_{|\partial_0 p|}\partial_0 p = \partial_{|\partial_1 p|}\partial_1 p$. The first globular condition guarantees that $(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p)$ is well-defined by the properties of (Whiskering). Together, the globular conditions also allow us to apply (Stable Boundaries) to deduce

$$\partial_{|\partial_0 p|}(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_0 p) = \partial_{|\partial_1 p|}(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p)$$

Thus, by the properties of (Concatenation) and since $(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_0 p)$ can be (post-)concatenated with $\left(\partial_{|d|}d\right)_R$ so can $(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p)$, showing the second concatenation '.' is defined.

On the other hand, the definedness of (pre-)concatenation with $\left(\partial_{|d|}d\right)_L$ should be clear as $(\partial_l\partial_{|d|}d \underset{c'}{\blacktriangleright} \partial_1 p)$ still has the initial boundary $\partial_l\partial_{|d|}d$ by (Whiskering). Thus we checked that (1.19) is indeed a well-defined diagram. □

Claim 1.20. *The boundaries of diagrams also satisfy the following globular condition, that is they satisfy the equations*

$$\partial_0\partial_0 d = \partial_0\partial_{|d|}d$$
$$\partial_{|\partial_0 d|}\partial_0 d = \partial_{\left|\partial_{|d|}d\right|}\partial_{|d|}d$$

Proof. In fact, we have

$$\partial_0\partial_0 d = \partial_0\partial_i d$$
$$\partial_{|\partial_0 d|}\partial_0 d = \partial_{|\partial_i d|}\partial_i d$$

and this can be seen inductively from the inductive definition of $\partial_i d$ (it is true for $i = 0$): When defining $\partial_{i+1} d$ the initial and final boundaries of $\partial_i d$ carry over to $\partial_{i+1} d$: Indeed, by (1.17) the latter is a (Concatenation) of $(k-1)$-diagrams of the form

$$(\partial_i d')_L . (\partial_l \partial_i d' \underset{c'}{\blacktriangleright} \partial_1 p).(\partial_i d')_R$$

while the former equals

$$(\partial_i d')_L . (\partial_l \partial_i d' \underset{c'}{\blacktriangleright} \partial_0 p).(\partial_i d')_R$$

Here, we defined d' to consist of the first i processes of d

$$d' := \mathrm{id}(\partial_0 d) \underset{c_1 d}{\triangleright} p_1 d \triangleright \ldots \underset{c_i d}{\triangleright} p_i d$$

The initial and final boundary of $\partial_{i+1} d$ and $\partial_i d$ thus coincide by (Concatenation), and by induction they coincide with the initial and final boundary of $\partial_0 d$. □

(iii)b k**-Diagram identities:** As before we introduce no non-trivial identity paths on D_n. But before we proceed we claim the following.

Claim 1.21.

(a) For $d, d_1, d_2 \in D_k$ we have that firstly,

$$\begin{aligned} & d_1 = d_2 \\ & \quad \iff \forall i.(\partial_i d_1, p_i d_1, c_i d_1) = (\partial_i d_2, p_i d_2, c_i d_2) \end{aligned}$$

Secondly, we have

$$\begin{aligned} & \mathrm{supp}(\lambda(i : \mathbb{N}).p_i d) = \mathrm{supp}(\lambda(i : \mathbb{N}).c_i d) = \{ \, i \mid 1 \leq i \leq |d| \, \} \\ & \mathrm{supp}(\lambda(i : \mathbb{N}).\partial_i d) = \{ \, i \mid 0 \leq i \leq |d| \, \} \end{aligned}$$

Consequently, we can write

$$d \equiv \ \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \ \cdots \ \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d \ \in D_k$$

in order to refer to a diagram d in D_k.

(b) Let e be a sequence

$$e = \ b_0 \xrightarrow[c_1]{p_1} b_1 \xrightarrow[c_2]{p_2} b_2 \ \cdots \ b_{s-1} \xrightarrow[c_s]{p_s} b_s$$

where $b_i \in (D_{k-1})_\perp$, $p_i \in C_k$ and $c_i \in \left(\mathbb{N}^{2k}\right)_\perp$. Then e corresponds (in the sense of part (a)) to a valid diagram $d \in D_k$ if and only if the following 'boundary conditions' conditions are fulfilled

(1.22)

$$\begin{aligned} \forall i \in \{ \, 1, ..., s \, \} \ . \Big(\ & b_{i-1}|_{c_i} = \partial_0 p_i \\ & \wedge b_i = (b_{i-1})_L . (\partial_l b_{i-1} \underset{c'_i}{\blacktriangleright} \partial_1 p_i). (b_{i-1})_R \ , \ \text{where } c'_i, (b_{i-1})_R, (b_{i-1})_L \text{ satisfy} \\ & c_i = (c'_i, l_i, r_i) \in \mathbb{N}^{2k} \ , \ |(b_{i-1})_L| = l_i \ , \ |(b_{i-1})_R| = r_i \quad \Big) \end{aligned}$$

Namely, under these conditions we can take

$$d = \mathrm{id}(b_0) \underset{c_1}{\triangleright} p_1 \underset{c_2}{\triangleright} p_2 ... \underset{c_s}{\triangleright} p_s$$

Remark 1.23. In case b_i, c_i are defined and boundary conditions (1.22) are met for e we identify e with d as an element of D_k. Otherwise we identify e with $\perp \in (D_k)_\perp$. In either case we see that $e \in (D_k)_\perp$.

We also emphasise that by properties of (Restriction) the boundary conditions have the following equivalent formulation

$$\forall i \in \{\, 1, ..., s \,\} \;.\; \Big(b_{i-1} = (b_{i-1})_L \,.(\partial_l b_{i-1} \underset{c'_i}{\blacktriangleright} \partial_0 p_i).\, (b_{i-1})_R \quad \wedge\, b_i = (b_{i-1})_L \,.(\partial_l b_{i-1} \underset{c'_i}{\blacktriangleright} \partial_1 p_i).\, (b_{i-1})_R \quad \text{where } c_i = (c'_i, l_i, r_i) \in \mathbb{N}^{2k} \;,\; |(b_{i-1})_L| = l_i \;,\; |(b_{i-1})_R| = r_i \Big) \tag{1.24}$$

♢

Proof. part (a): *Both statements follow by a standard inductive argument which the reader is invited to skip.* The second statement follows by structural induction on d and the inductive definitions of ∂_i, p_i and c_i. For instance, in the case of c_i the statement holds if $d = \mathrm{id}(b)$, as then $\mathrm{supp}\,(\lambda(i:\mathbb{N}).c_i(\mathrm{id}(b))) = \emptyset$. For $d = d' \underset{c}{\triangleright} p$, we have by induction $\mathrm{supp}\,(\lambda(i:\mathbb{N}).c_i(d'))) = \{\, i \;\mid\; 1 \leq i \leq |d'| \,\}$. But $|d| := |d'| + 1$ and $c_{|d'|+1}(d) := c \neq \bot$ while $c_{i>|d'|+1} := c_i(d') = \bot$ and so the statement follows. Similar arguments hold for ∂_i and p_i.

Both directions of the first statement follow by mutual structural induction on d_1, d_2. If $d_i = \mathrm{id}(b_i), i = 1, 2$, then for all j, $c_j d_i = p_j d_i = \bot$, and d_1 matches d_2 iff b_1 and b_2 match iff $\partial_j d_1 = \partial_j d_2$ for all j. If one of d_1, d_2 equals $\mathrm{id}(b)$ and the other equals $d' \underset{c}{\triangleright} p$ then d_1, d_2 do not match but neither do ∂_j, c_j, p_j on them as they have different supports by the second statement of part (a) which we just proved. If finally $d_i = d'_i \underset{c_i}{\triangleright} p_i, i = 1, 2$, then $d_1 = d_2$ iff $d'_1 = d'_2$, $c_1 = c_2$ and $p_1 = p_2$ iff (by inductive hypothesis for d'_i and definition of ∂_j, c_j and p_j) the right hand side of the statement holds.

part (b): For the 'only if' direction we assume that e is the sequence of some diagram d. By part (a) we have

$$b_i = \partial_i d \in D_k \;,\; 0 \leq i \leq s = |d|$$
$$c_i = c_i d \in \mathbb{N}^{2k} \;,\; 1 \leq i \leq s = |d|$$

d itself is a diagram of the form

$$d = \mathrm{id}(\partial_0 d) \underset{c_1 d}{\triangleright} p_1 d \underset{c_2 d}{\triangleright} p_2 d \;\triangleright\; \ldots \underset{c_{|d|} d}{\triangleright} p_{|d|} d$$

and thus the first boundary condition $b_{i-1}|_{c_i} = \partial_0 p_i$ in (1.22) follows from the typing conditions of constructors $\triangleright$ in d, while the second boundary condition $b_i = (b_{i-1})_{l_i} .(\partial_l b_{i-1} \underset{c}{\blacktriangleright} \partial_1 p).\, (b_{i-1})_{r_i}$ in (1.22) follows from the inductive definition of ∂_i.

For the 'if' direction we need to show that the candidate diagram

$$d = \mathrm{id}(b_0) \underset{c_1}{\triangleright} p_1 \underset{c_2}{\triangleright} p_2 ... \underset{c_s}{\triangleright} p_s$$

is indeed a valid diagram by exhibiting witnesses of the equalities

$$\Big(\partial_i \underbrace{\Big(\mathrm{id}(b_0) \underset{c_1}{\triangleright} p_1 \underset{c_2}{\triangleright} p_2 ... \underset{c_i}{\triangleright} p_i\Big)}_{=:d_i} \Big)\Big|_{c_i} = \partial_0 p_{i+1}$$

Under the inductive assumption (which is true for $i = 1$) that

$$b_{i-1} \overset{!}{=} \partial_{i-1}(d_{i-1}) := \partial_{i-1} \Big(\mathrm{id}(b_0) \underset{c_1}{\triangleright} p_1 \underset{c_2}{\triangleright} p_2 ... \underset{c_{i-1}}{\triangleright} p_{i-1}\Big)$$

the first boundary condition from (1.22) guarantees that $\partial_{i-1}(d_{i-1})|_{c_{i-1}} = \partial_0 p_i$, i.e. we can glue p_i to d_{i-1}. By the definition of ∂_i and the second condition $b_i = (b_{i-1})_{l_i} .(\partial_l b_{i-1} \underset{c}{\blacktriangleright} \partial_1 p).\, (b_{i-1})_{r_i}$ this yields

$$b_i = \partial_i \Big(\mathrm{id}(b_0) \underset{c_1}{\triangleright} p_1 \underset{c_2}{\triangleright} p_2 ... \underset{c_i}{\triangleright} p_i\Big) = \partial_i(d_i)$$

In this way we can inductively construct the required witnesses. □

(iv) **k-Diagram operations** (1.4)**:** Using Remark 1.23 to identify diagrams with their respective sequences, we can define concatenation and restriction directly without resorting to pattern matching.

We start with the definition of ***concatenation***

$$-.- : D_k \times D_k \to (D_k)_\perp \tag{1.25}$$

$$d, d' \mapsto \begin{cases} \partial_0 d \xrightarrow[c_1 d]{p_1 d} \cdots \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d \xrightarrow[c_1 d']{p_1 d'} \cdots \xrightarrow[c_{|d'|} d']{p_{|d'|} d'} \partial_{|d'|} d' & \text{if } \partial_{|d|} d = \partial_0 d' \\ \perp & \text{otherwise} \end{cases}$$

Claim 1.26. *For $d, d' \in D_n$, $d.d' \in (D_k)_\perp$ is defined if and only if $\partial_{|d|} d = \partial_0 d'$.*

Proof. If $\partial_{|d|} d \neq \partial_0 d'$ then $d.d'$ is certainly undefined. On the other hand, if $\partial_{|d|} d = \partial_0 d'$ then the given sequence for $d.d'$ is indeed a valid diagram since it satisfies the boundary conditions from Claim 1.21 (b). □

Next we define ***restriction*** of k-diagrams at coordinates $c = (c', l, r) \in \mathbb{N}^{2(k+1)}$ which uses the definition of restriction in lower dimensions

(1.27)

$$-|_{(c',l,r)} : D_k \to (D_k)_\perp$$

$$d \mapsto \begin{cases} (\partial_l d)|_{c'} \xrightarrow[(c_{l+1} d) - c']{p_{l+1} d} (\partial_{l+1} d)|_{c'} \to \cdots \to \left(\partial_{|d|-r+1} d\right)\big|_{c'} \xrightarrow[(c_{|d|-r} d) - c']{p_{|d|-r} d} \left(\partial_{|d|-r} d\right)\big|_{c'} & \text{if } l + r \leq |d| \\ \perp & \text{otherwise} \end{cases}$$

We remark (in the first case for $l+r \leq |d|$) that some of the boundaries $(\partial_j d)|_{c'}$ or coordinates $(c_j d) - c'$ (cf. Remark 1.13) of the above sequence might be undefined, but that this is accounted for in Claim 1.21 (b) and Remark 1.23 : In this case the sequence is identified with the undefined diagram $\perp$ in $(D_k)_\perp$.

Finally, we give a definition of ***whiskering*** at coordinates $c = (c', l, r) \in \mathbb{N}^{2(k+1)}$. The definition is quite intuitive and straightforward, however showing definedness will take up a bit more space.

$$- \underset{c}{\blacktriangleright} - : D_{k-1} \times D_k \to (D_k)_\perp$$

$$b, d \mapsto \begin{cases} \mathrm{id}(b) \underset{c_1 d + c}{\triangleright} p_1 d \underset{c_2 d + c}{\triangleright} \ldots \triangleright p_{|d|-1} d \underset{c_{|d|} d + c}{\triangleright} p_{|d|} d & \text{if } b|_c = \partial_0 d \\ \perp & \text{otherwise} \end{cases}$$

In the first case (which assumes $b|_c = \partial_0 d$) we need to provide witnesses for the $\triangleright$-constructors' side conditions to guarantee their applicability. For given b, d and assuming $b|_c = \partial_0 d$, these witnesses are constructed inductively as follows:

Construction 1.28. We set $c = (c', l, r)$, $c_i d = c_i = (c'_i, l_i, r_i)$ and $p_i d =: p_i$. Our inductive assumption is

$$\partial_i \Big(\underbrace{\mathrm{id}(b) \underset{c_1 + c}{\triangleright} p_1 \triangleright \ldots \underset{c_i + c}{\triangleright} p_i}_{=:\tilde{d}_i} \Big) = b_L . \left(\partial_l b \underset{c'}{\blacktriangleright} \partial_i d \right) . b_R \tag{1.29}$$

where b_R, b_L are such that $|b_L| = l, |b_R| = r$. This is true for $i = 0$ by our hypothesis $b|_c = \partial_0 d$ together with an application of (Restriction). However, d also satisfies the following boundary condition (1.22)

$$\partial_i d\big|_{c_{i+1}} = \partial_0 p_{i+1}$$

and thus by again by (Restriction)

$$\partial_i d = (\partial_i d)_{L,i} \cdot \left(\partial_{l_{i+1}} \partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) . (\partial_i d)_{R,i} \tag{1.30}$$

with diagrams $(\partial_i d)_{L,i}, (\partial_i d)_{R,i}$ satisfying $\left|(\partial_i d)_{L,i}\right| = l_{i+1}$ and $\left|(\partial_i d)_{R,i}\right| = r_{i+1}$. Substituting (1.30) for $\partial_i d$ in (1.29) we obtain

(1.31)

$$\partial_i \tilde{d}_i = b_L . \left(\partial_l b \underset{c'}{\blacktriangleright} \partial_i d \right) . b_R$$

$$= b_L . \left(\partial_l b \underset{c'}{\blacktriangleright} \left((\partial_i d)_{L,i} . \left(\partial_{l_{i+1}} \partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) . (\partial_i d)_{R,i} \right) \right) . b_R$$

$$= \underbrace{b_L . \left(\partial_l b \underset{c'}{\blacktriangleright} (\partial_i d)_{L,i} \right)}_{b_{L,i}} . \left(\partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'}{\blacktriangleright} \left(\partial_{l_{i+1}} \partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) \right) . \underbrace{\left(\partial_{|\partial_i \tilde{d}_i| - r - r_{i+1}} \partial_i \tilde{d}_i \underset{c'}{\blacktriangleright} (\partial_i d)_{R,i} \right) . b_R}_{b_{R,i}}$$

$$= b_{L,i} . \left(\partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'}{\blacktriangleright} \left(\partial_{l_{i+1}} \partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) \right) . b_{R,i}$$

In the first step we performed the substitution, in the second step we applied (Distributivity) twice to distribute the three coloured terms, in the last step we condensed notation. Note that (by properties of (Whiskering) and (Concatenation)) we have

$$\begin{aligned} |b_{L,i}| &= l + l_{i+1} \\ |b_{R,i}| &= r + r_{i+1} \end{aligned} \tag{1.32}$$

Further, from the sequence representation of (Whiskering) (applied here in lower dimensions than k) we can read off the following equality

$$\partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'}{\blacktriangleright} \left(\partial_{l_{i+1}} \partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) = \partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'+c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \tag{1.33}$$

Substituting (1.33) into the expression for $\partial_i \tilde{d}_i$ obtained in (1.31) we get

$$\partial_i \tilde{d}_i = b_{L,i} . \left(\partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'+c'_{i+1}}{\blacktriangleright} \partial_0 p_{i+1} \right) . b_{R,i} \tag{1.34}$$

Now (1.34) and (1.32) together imply that the side condition for the constructor application

$$\tilde{d}_{i+1} := \tilde{d}_i \underset{c_{i+1}+c}{\triangleright} p_{i+1}$$

is satisfied. From our inductive assumption (1.29) on $\tilde{d}_i$ we deduce that

$$\tilde{d}_{i+1} = \mathrm{id}(b) \underset{c_1+c}{\triangleright} p_1 \triangleright \dots \underset{c_i+c}{\triangleright} p_i \underset{c_{i+1}+c}{\triangleright} p_{i+1}$$

By definition of ∂_{i+1} we have

$$\partial_{i+1} \tilde{d}_{i+1} = b_{L,i} . \left(\partial_{l+l_{i+1}} \partial_i \tilde{d}_i \underset{c'+c'_{i+1}}{\blacktriangleright} \partial_1 p_{i+1} \right) . b_{R,i}$$

But from here we can now trace backwards the equality (1.33) and the last two steps in (1.31) to arrive at

$$\partial_{i+1}\tilde{d}_{i+1} = b_L.\left(\partial_l b \underset{c'}{\blacktriangleright} \left((\partial_i d)_{L,i}.\left(\partial_{l_{i+1}}\partial_i d \underset{c'_{i+1}}{\blacktriangleright} \partial_1 p_{i+1}\right).(\partial_i d)_{R,i}\right)\right).b_R$$

Again by definition of ∂_{i+1} (this time on d) the right hand side can then be collapsed to

$$\partial_{i+1}\tilde{d}_{i+1} = b_L.\left(\partial_l b \underset{c'}{\blacktriangleright} \partial_{i+1} d\right).b_R$$

Comparing this to (1.29) we see that we have completed the induction step from i to $i+1$. As a consequence, each application of a constructor in

$$\mathrm{id}(b) \underset{c_1 d+c}{\triangleright} p_1 d \underset{c_2 d+c}{\triangleright} \dots \triangleright p_{|d|-1} d \underset{c_{|d|} d+c}{\triangleright} p_{|d|} d$$

is valid under our initial assumption $b|_c = \partial_0 d$. Thus whiskering of d on b at c is defined whenever $b|_c = \partial_0 d$. □

We have just proven part (a) of the following claim.

Claim 1.35. *Concatenation, restriction and whiskering as defined in this section satisfy the following.*

(a) (Whiskering) *Given* $d = \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \cdots \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_s d} \partial_{|d|} d \in D_k$, *and* $b \in D_{k-1}$, *then* $b \underset{c}{\blacktriangleright} d$ *is defined if and only if* $b|_c = \partial_0 d$ *and in this case we have:*

$$b \underset{c}{\blacktriangleright} d = b \xrightarrow[c+c_1 d]{p_1 d} b_1 \xrightarrow[c+c_2 d]{p_2 d} b_2 \cdots b_{|d|-1} \xrightarrow[c+c_{|d|} d]{p_{|d|} d} b_{|d|}$$

where $b_i = \partial_i(b \underset{c}{\blacktriangleright} d)$, $i \geq 1$, *and* $c + c_i$ *denotes vector addition (cf. Remark 1.13).*

(b) (Concatenation) *For* $d, d' \in D_k$, $d.d'$ *is defined if and only if* $\partial_{|d|} d = \partial_0 d'$ *and in this case we have:*

$$d.d' = \partial_0 d \xrightarrow[c_1 d]{p_1 d} \cdots \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d \xrightarrow[c_1 d']{p_1 d'} \cdots \xrightarrow[c_{|d'|} d']{p_{|d'|} d'} \partial_{|d'|} d'$$

Note that $(-.-)$ *is associative and thus we can write e.g.* $d.d'.d''$ *without brackets*

(c) (Restriction) *For* $d, d' \in D_k$ *we have*

$$d|_{(c,l,r)} = d' \iff d = d_L.(\partial_l d \underset{c}{\blacktriangleright} d').d_R \ , \ \text{where } |d_L| = l, |d_R| = r$$

(d) (Distributivity) *For* $d, d' \in D_k$, $b \in D_{k-1}$ *we have*

$$b \underset{c}{\blacktriangleright} (d.d') = \left(b \underset{c}{\blacktriangleright} d\right).\left(b' \underset{c}{\blacktriangleright} d'\right)$$

where $b' = \partial_{|d|}(b \underset{c}{\blacktriangleright} d)$.

(e) (Stable Boundaries) *For* $d, d' \in D_k$ *with* $\partial_0 d = \partial_0 d'$ *and* $\partial_{|d|} d = \partial_{|d'|} d'$ *and* $b \in D_{n-1}$ *we have*

$$\partial_{|d|}(b \underset{c}{\blacktriangleright} d) = \partial_{|d'|}(b \underset{c}{\blacktriangleright} d')$$

in case either the left or the right hand side is defined.

Proof. Part (a) was proven in Construction 1.28. Part (b) is just restating the definition (1.25). With part (a) and (b) at hand, parts (d) and (e) can both be proven from Construction 1.28 by inspection of the inductive assumption (1.29). More explicitly, for (d) the claim (1.29) implies that

$$b' := \partial_{|d|}(b \underset{c}{\blacktriangleright} d) = b_L.(\partial_l b \underset{c'}{\blacktriangleright} \partial_{|d|} d).b_R$$

where we set $c = (c', l, r)$, and $|b_L| = l, |b_R| = r$. Assuming $b \underset{c}{\blacktriangleright} d$ is defined, we thus see that $(b' \underset{c}{\blacktriangleright} d')$ is defined iff $d.d'$ is defined, namely iff $\partial_{|d|} d = b'|_c = \partial_0 d'$. Thus both sides are defined iff $b|_c = \partial_0 d$ and $\partial_{|d|} d = \partial_0 d'$. In this case, equality of the left hand side and right hand side follows from comparing processes, coordinates (using (a)) and the first boundary. For part (e) on the other hand, note that both sides are defined iff $b|_c = \partial_0 d = \partial_0 d'$. In this case, equality follows since by claim (1.29) the left hand side equals

$$\partial_{|d|}(b \underset{c}{\blacktriangleright} d) = b_L.(\partial_l b \underset{c'}{\blacktriangleright} \partial_{|d|} d).b_R$$

where we set $c = (c', l, r)$, and $|b_L| = l, |b_R| = r$. With the same definitions, the right hand side equals

$$\partial_{|d'|}(b \underset{c}{\blacktriangleright} d) = b_L.(\partial_l b \underset{c'}{\blacktriangleright} \partial_{|d'|} d').b_R$$

which by assumption $\partial_{|d|} d = \partial_{|d'|} d'$ implies the statement of (e).

We are left with proving part (c). Writing out the definition (1.27) for $d|_{(c,l,r)}$, we need to proof the following

$$(\partial_l d)|_c \xrightarrow[(c_{l+1}d)-c]{p_{l+1}d} \cdots \xrightarrow[(c_{|d|-r}d)-c]{p_{|d|-r}d} \left(\partial_{|d|-r} d\right)\Big|_c = d'$$
$$\iff d = d_L.(\partial_l d \underset{c}{\blacktriangleright} d').d_R \ , \text{ where } |d_L| = l, |d_R| = r$$

<u>For the $\Rightarrow$ direction</u> we assume the left hand side of the mutual implication holds. Note that d' is defined and thus $d|_{(c,l,r)}$ needs to be defined too. The hypothesis $d|_{(c,lr)} = d'$ implies equality of the first boundary $(\partial_l d)|_c = \partial_0 d'$ and thus $(\partial_l d \underset{c}{\blacktriangleright} d')$ is defined by part (a). Further, by hypothesis on d' and part (a) we obtain

$$(\partial_l d \underset{c}{\blacktriangleright} d') = \partial_l d \xrightarrow[c+(c_{l+1}d-c)]{p_{l+1}d} (\partial_l d)_1 \xrightarrow[c+(c_{l+2}d-c)]{p_{l+2}d} (\partial_l d)_2 \ \cdots \ \partial_{|d|-r-1} d \xrightarrow[c+(c_{|d|-r}d-c)]{p_{|d|-r}d} (\partial_l d)_{|d'|}$$
$$= \partial_l d \xrightarrow[c_{l+1}d]{p_{l+1}d} \partial_{l+1} d \xrightarrow[c_{l+2}d]{p_{l+2}d} \partial_{l+2} d \ \cdots \ \partial_{|d|-r-1} d \xrightarrow[c_{|d|-r}d]{p_{|d|-r}d} \partial_{|d|-r} d$$

where we not only computed $c + (c_{l+i}d - c) = c_{l+i}d$ but also equated $(\partial_l d)_i = \partial_{l+i} d$: To see why this can be done we first note both sequences are valid: The first sequences was derived by part (a) satisfying (a)'s definedness condition. The second one is valid because it is a subsequence of d, and thus fulfills the boundary conditions (1.22). But then, having the same initial boundary as well as the same processes and coordinates both sequences determine the same diagram (cf. Claim 1.21) and thus can be equated. Setting

$$d_L = \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \ldots \xrightarrow[c_l d]{p_l d} \partial_l d \ \in D_k$$
$$d_R = \partial_{|d|-r} d \xrightarrow[c_{|d|-r+1}d]{p_{|d|-r+1}d} \partial_{|d|-r+1} d \ldots \xrightarrow[c_{|d|}d]{p_{|d|}d} \partial_{|d|} d \ \in D_k$$

by part (b) (describing concatenation) we obtain $d = d_L.(\partial_l d \underset{c}{\blacktriangleright} d').d_R$ as required. Also note that $|d_L| = l$, $|d_R| = r$ as they are described by sequences of length l and r respectively.

<u>For the $\Leftarrow$ direction</u> we assume the right hand side of the mutual implication holds. First note that d is defined and so we must have $l + r \leq |d|$ and $\partial_l d|_c = \partial_0 d'$. Further, setting

$$d' = \partial_0 d' \xrightarrow[c_1 d']{p_1 d'} \partial_1 d' \ldots \xrightarrow[c_{|d'|}d']{p_{|d'|}d'} \partial_{|d'|} d' \ \in D_k$$

we can apply part (a) yielding

$$(\partial_l d \underset{c}{\blacktriangleright} d') = \ \partial_l d \xrightarrow[c+c_1 d']{p_1 d'} (\partial_l d)_1 \xrightarrow[c+c_2 d']{p_2 d'} (\partial_l d)_2 \ \cdots \ (\partial_l d)_{|d'|-1} \xrightarrow[c+c_{|d'|} d']{p_{|d'|} d'} (\partial_l d)_{|d'|}$$

However, since $d_L.(\partial_l d \underset{c}{\blacktriangleright} d').d_R = d$ and applying part (b) we also have

$$(\partial_l d \underset{c}{\blacktriangleright} d') = \ \partial_l d \xrightarrow[c_{l+1} d]{p_{l+1} d} \partial_{l+1} d \xrightarrow[c_{l+2} d]{p_{l+2} d} \partial_{l+2} d \ \cdots \ \partial_{l+|d'|-1} d \xrightarrow[c_{l+|d'|} d]{p_{l+|d'|} d} \partial_{l+|d'|} d$$

And thus we can equate for

$$\begin{aligned} (\partial_l d)_i &= \partial_{l+i} d \qquad &, \ 0 \leq i \leq |d'| \\ p_i d' &= p_{l+i} d \qquad &, \ 1 \leq i \leq |d'| \\ c + c_i d' &= c_{l+i} d \qquad &, \ 1 \leq i \leq |d'| \end{aligned}$$

and thus

$$c_i d' = (c_{l+i} d) - c \quad , \ 1 \leq i \leq |d'|$$

Together with the previous observation that $\partial_l d|_c = \partial_0 d'$, we deduce as required

$$(\partial_l d)|_c \xrightarrow[(c_{l+1} d)-c]{p_{l+1} d} \cdots \xrightarrow[(c_{|d|-r} d)-c]{p_{|d|-r} d} \left(\partial_{|d|-r} d\right)\big|_c \ = d'$$

because any diagram is uniquely determined by its processes, process coordinates and a single boundary (which either follows by definition of ∂_i or, for the present case of the initial boundary it can also be seen from Claim 1.21). □

Lemma 1.36. *The Inductive Claim 1.10 holds for $n = k$.*

Proof. The required proofs have been done in Claim 1.15, Claim 1.21 and Claim 1.35.

2. Lists with duals

As explained in the introduction, the previous section describes the 'core' datatype to capture the behaviour of higher lists. We will now extend this behaviour in three usefule ways: Namely to include ***Compositionality*** of lists ('higher lists of list'), ***Duality*** of list elements (which means elements can be both resources/values or co-resources/continuations) and ***Unitarity*** which coinductively describes when a resource and co-resources *compose* to the identity.

The discussion of the first two of these extension will follow the same scheme: We extend the set of constructors for C_n, define ∂_0, ∂_1 on these extensions and show that they satisfy the globular condition. This is the required extension of Claim 1.15, and in fact the only required addition to the inductive step Lemma 1.36.

Before we proceed we introduce the following useful definition and notational convention.

Definition 2.1 (Segments)**.** For a diagram

$$d = \left(b_0 \xrightarrow[c_1]{p_1} b_1 \to \cdots \to b_{k-1} \xrightarrow[c_k]{p_k} b_k \right) \in D_n$$

we define, for $0 \leq i \leq j \leq k$, the $[i,j]$-*segment of* d, denoted by $d_{[i,j]}$, as follows

$$d_{[i,j]} = \left(b_i \xrightarrow[c_{i+1}]{p_{i+1}} b_{i+1} \to \cdots \to b_{j-1} \xrightarrow[c_j]{p_j} b_j \right)$$

As a shorthand we set $d_{[i]} := d_{[i,i+1]}$, and refer to it as the ith segment of d. We also emphasise that $d_{[i,i]} = \mathrm{id}(\partial_i d) = \mathrm{id}(b_i)$. ◆

Notation 2.2 (Processes are size 1 diagrams)**.** For any $p \in C_n$, $\mathrm{id}(\partial_0 p) \triangleright p$ is a defined diagram of size 1 since the side condition for application of $\triangleright$ is manifestly fulfilled (recall that if coordinates are kept implicit by convention we assume them to be $0 \in \mathbb{N}^{2n}$). Conveniently, we have that $\partial_i(\mathrm{id}(\partial_0) \triangleright p) = \partial_i p, i = 0, 1$. By abuse of notation we then set

$$p \equiv \mathrm{id}(\partial_0 p) \triangleright p \in D_n$$

In this way, we can associate cells to diagrams of size 1. Since in sequence notation we could also write these diagrams as

$$\mathrm{id}(\partial_0 p) \triangleright p = \ \partial_0 p \xrightarrow{\ p\ } \partial_1 p$$

we will often use the notation

$$p :\ \partial_0 p \longrightarrow \partial_1 p$$

for a cell p. Moreover, $\blacktriangleright$ and $\triangleright$ collapse under this convention, in the sense that, whenever $b|_c = \partial_0 p$ we have

$$\mathrm{id}(b) \underset{c}{\triangleright} p = b \underset{c}{\blacktriangleright} p$$

since $\partial_0(-)$, $c_1(-)$, $p_1(-)$ coincide for the left and the right hand side (on the right hand side we force $p \in D_n$ by the identification that was just introduced). ♢

2.1. **Compositionality.**

For all $n \in \mathbb{N}$, composition of diagrams yields new cells which we denote by:

$$\begin{aligned} p : C_n :=& \quad \dots \\ & \mid \quad \langle d \rangle \ , \ d \in D_n \end{aligned}$$

These compositions are witnessed by higher cells, and thus we further extend C_{n+1} as follows

$$\begin{aligned} p : C_{n+1} :=& \quad \dots \\ & \mid \quad d \Rightarrow \langle d \rangle \ , \ d \in D_n \end{aligned}$$

Having introduced new cells, we now need to extend the boundary function (∂_0, ∂_1). For $\langle d \rangle$ we do so by setting

$$\begin{aligned} \partial_0 \langle d \rangle &= \partial_0 d \\ \partial_1 \langle d \rangle &= \partial_{|d|} d \end{aligned}$$

For $(d \Rightarrow \langle d \rangle)$ we use Notation 2.2 such that the following boundaries for witnesses of composition in C_{n+1} make sense:

$$\begin{aligned} \partial_0(d \Rightarrow \langle d \rangle) &= d \\ \partial_1(d \Rightarrow \langle d \rangle) &= \langle d \rangle \end{aligned} \tag{2.3}$$

Finally, we need to verify that the globular conditions for (1.15) stay true. After substituting the above definitions, in the case of $\langle d \rangle$ we need to verify the following (highlighted in blue)

$$\partial_0 \partial_0 \langle d \rangle := \partial_0 \partial_0 d \overset{!}{=} \partial_0 \partial_{|d|} d =: \partial_0 \partial_1 \langle d \rangle$$

$$\partial_{|\partial_0 \langle d \rangle|} \partial_0 \langle d \rangle := \partial_{|\partial_0 d|} \partial_0 d \overset{!}{=} \partial_{\left|\partial_{|d|} d\right|} \partial_{|d|} d =: \partial_{|\partial_1 \langle d \rangle|} \partial_1 \langle d \rangle$$

and in the case of $(d \Rightarrow \langle d \rangle)$ in C_{n+1} we need to verify

$$\partial_0 \partial_0 (d \Rightarrow \langle d \rangle) := \partial_0 d \overset{!}{=} \partial_0 \langle d \rangle =: \partial_0 \partial_1 (d \Rightarrow \langle d \rangle)$$

$$\partial_{|\partial_0 (d \Rightarrow \langle d \rangle)|} \partial_0 (d \Rightarrow \langle d \rangle) := \partial_{|d|} d \overset{!}{=} \partial_1 \langle d \rangle =: \partial_{|\partial_1 (d \Rightarrow \langle d \rangle)|} \partial_1 (d \Rightarrow \langle d \rangle)$$

Now, the first two equations follow from Claim 1.20. The last two equations are true by definition (2.3).

Remark 2.4 (Composition and identities). For later use we introduce the shorthand

$$1(d) := \langle \mathrm{id}(d) \rangle$$

$\diamondsuit$

2.2. **Duality.**

Dualisation yields new cells for all $n \in \mathbb{N}$ as follows:

$$\begin{aligned} p : C_n := \quad & \dots \\ | \quad & p^* \ , \ p \in C_n \end{aligned}$$

These duals (or 'deficits' as discussed in the introduction) are introduced and eliminated by higher cells called *shifts*. Since elimination will in fact be dual to introduction itself, we need to extend C_{n+1} only with the (two) shift cells for, say, elimination as follows

$$\begin{aligned} p : C_{n+1} := \quad & \dots \\ | \quad & \uparrow_R(p) \ , \ p \in C_n \\ | \quad & \uparrow_L(p) \ , \ p \in C_n \end{aligned}$$

We will refer to $\uparrow_R(p)$ as *right elimination*, $\uparrow_L(p)$ as *left elimination* and introduce the shorthands $\downarrow_R(p) := (\uparrow_R(p))^*$, $\downarrow_L(p) := (\uparrow_L(p))^*$ referred to as *right introduction* and *left introduction* respectively.

As for compositionality we need to extend the boundary maps onto these new cells and we do so as follows

$$\begin{aligned} \partial_0 p^* &= \partial_1 p \\ \partial_1 p^* &= \partial_0 p \end{aligned} \tag{2.5}$$

From this definition we see that assuming inductively p satisfies the globular conditions, it follows that p^* satisfies the globular conditions for Claim 1.15 as well.

For shifts we then define

$$\begin{aligned} \partial_0 \uparrow_R(p) &= p.p^* \\ \partial_1 \uparrow_R(p) &= \mathrm{id}(\partial_0 p) \\ \partial_0 \uparrow_L(p) &= p^*.p \\ \partial_1 \uparrow_L(p) &= \mathrm{id}(\partial_1 p) \end{aligned}$$

Note that e.g. $p.p*$ is (by Notation 2.2 and (Concatenation)) the following diagram

$$p.p^* = \ \partial_0 p \xrightarrow[0]{p} \partial_1 p \xrightarrow[0]{p^*} \partial_0 p$$

This has the same initial and final boundary as $\mathrm{id}(\partial_0 p)$ and thus the globular conditions are fulfilled for $\uparrow_R(p)$ and Claim 1.15 can be extended accordingly. The same argument shows this also holds for $\uparrow_L(p)$.

We now extend dualisation and shifts (which so far only act on cells) to diagrams:

(i) Dualisation on diagrams $(-)^* : D_n \to D_n$ is defined by

$$d = \partial_0 d \xrightarrow[c_1 d]{p_1 d} \partial_1 d \xrightarrow[c_2 d]{p_2 d} \partial_2 d \ \cdots \ \partial_{|d|-1} d \xrightarrow[c_{|d|} d]{p_{|d|} d} \partial_{|d|} d$$

$$\mapsto d^* = \ \partial_{|d|} d \xrightarrow[c_{|d|} d]{(p_{|d|} d)^*} \partial_{|d|-1} d \xrightarrow[c_{|d|-1} d]{(p_{|d|-1} d)^*} \partial_{|d|-2} d \ \cdots \ \partial_1 d \xrightarrow[c_1 d]{(p_1 d)^*} \partial_0 d$$

We should show that the sequence given for the image of d indeed describes a valid diagram (in the sense of Claim 1.21). Note that $d \in D_n$ implies that

$$\partial_{i-1} d \xrightarrow[c_i d]{p_i d} \partial_i d$$

is a valid diagram. In particular, setting $c_i d = (c'_i, l_i, r_i)$, we have by the boundary conditions (1.24)

$$\begin{aligned}
\partial_{i-1} d &= b_{i,L}.\left(\partial_{l_i}\partial_{i-1} d \underset{c'_i}{\blacktriangleright} \partial_0 p_i d\right).b_{i,R} \\
&= b_{i,L}.\left(\partial_{l_i}\partial_{i-1} d \underset{c'_i}{\blacktriangleright} \partial_1 (p_i d)^*\right).b_{i,R} \quad \text{where } |b_{i,L}| = l_i \ , \ |b_{i,R}| = r_i \\
\partial_i d &= b_{i,L}.\left(\partial_{l_i}\partial_{i-1} d \underset{c'_i}{\blacktriangleright} \partial_1 p_i d\right).b_{i,R} \\
&= b_{i,L}.\left(\partial_{l_i}\partial_{i-1} d \underset{c'_i}{\blacktriangleright} \partial_0 (p_i d)^*\right).b_{i,R} \quad \text{where } |b_{i,L}| = l_i \ , \ |b_{i,R}| = r_i
\end{aligned}$$

for some diagrams $b_{i,L}, b_{i,R}$. Here we used the definition of $(p_i d)^*$ and its boundaries (2.5). It follows that the sequence

$$\partial_i d \xrightarrow[c_i d]{(p_i d)^*} \partial_{i-1} d \tag{2.6}$$

satisfies the boundary conditions (1.24) and is also a valid diagram. Thus the defining sequence of d^* given above, which is a (Concatenation) of the subsequences (2.6), is a valid diagram as well.

(ii) The (right eliminating) shift for a diagram $d \in D_n$, denoted by $\uparrow_R(d)$, can be defined quite elegantly by induction on the size of d, $k = |d|$. However, we remark this is not the unique way in which the cell $\uparrow_R(d)$ could be defined (cf. Remark 2.9).

Construction 2.7. We claim inductively in $k = |d|$, that for all $d \in D_n$, $|d| \geq 1$, we have a cell

$$\uparrow_R(d) : d.d^* \to \mathrm{id}(\partial_0 d)$$

Proof. For $k = 1$, we can take

$$d = (b_0 \underset{c}{\blacktriangleright} p)$$

where $b_i = \partial_i d$, and thus

$$d^* = (b_1 \underset{c}{\blacktriangleright} p^*)$$

Using (Distributivity) we find

$$\begin{aligned}
d.d^* &= (b_0 \underset{c}{\blacktriangleright} p).(b_1 \underset{c}{\blacktriangleright} p^*) \\
&= b_0 \underset{c}{\blacktriangleright} (p.p^*)
\end{aligned}$$

And thus we can set $\uparrow_R(d)$ to be the composite of the diagram

$$\begin{array}{c}
d.d* = b_0 \underset{c}{\blacktriangleright} (p.p^*) \\
{\scriptstyle (c,0,0)}\Big\downarrow{\scriptstyle \uparrow_R(p)} \\
\mathrm{id}(\partial_0 d) = b_0 \underset{c}{\blacktriangleright} (\mathrm{id}(\partial_0 p))
\end{array}$$

For general k, assuming $\uparrow_R(d')$ has been constructed for all d' with $|d'| < k$, we can take

$$d = (b_0 \underset{c_1}{\blacktriangleright} p_1).d_{[1,k]}$$

where $b_i = \partial_i d$, and $d_{[1,k]}$ denotes the $[1,k]$-segment of d which is of size $k-1$ (and was defined in Definition 2.1). We have

$$d^* = \left((b_0 \underset{c_1}{\blacktriangleright} p_1).d_{[1,k]} \right)^* = d^*_{[1,k]}.(b_1 \underset{c_1}{\blacktriangleright} p_1^*)$$

and we can then set $\uparrow_R (() \, d)$ to be the composite of the diagram

$$\begin{array}{c} d.d^* \\ = (b_0 \underset{c_1}{\blacktriangleright} p_1).d_{[1,k]}.d^*_{[1,k]}.(b_1 \underset{c_1}{\blacktriangleright} p_1^*) \\ {\scriptstyle (0,1,1)} \Big\downarrow {\scriptstyle \uparrow_R (d_{[1,k]})} \\ (b_0 \underset{c_1}{\blacktriangleright} p_1).\mathrm{id}(\partial_0 d_{[1,k]}).(b_1 \underset{c_1}{\blacktriangleright} p_1^*) \\ = (b_0 \underset{c_1}{\blacktriangleright} p_1).(b_1 \underset{c_1}{\blacktriangleright} p_1^*) \\ \overset{\text{(Dist)}}{=} (b_0 \underset{c_1}{\blacktriangleright} (p_1.p_1^*)) \\ {\scriptstyle (c_1,0,0)} \Big\downarrow {\scriptstyle \uparrow_R (p_1)} \\ (b_0 \underset{c_1}{\blacktriangleright} (\mathrm{id}(\partial_0 p))) \\ = \mathrm{id}(\partial_0 d) \end{array}$$

which completes the inductive construction of $\uparrow_R (d)$. ♢

Definition 2.8. In the same inductive fashion, but using the other remaining shifts

$$\uparrow_L (p) \, , \downarrow_R (p) \, , \downarrow_L (p)$$

instead of $\uparrow_R (p)$, we can define, for $d \in D_n$, $|d| \geq 1$, the following $(n+1)$-cells

$$\begin{aligned} \uparrow_L (d) &: d^*.d \to \mathrm{id}(\partial_{|d|} d) \\ \downarrow_R (d) &: \mathrm{id}(\partial_0 d) \to d.d^* \\ \downarrow_L (d) &: \mathrm{id}(\partial_{|d|} d) \to d^*.d \end{aligned}$$

For all of their constructions no new difficulties arise, and thus we can safely leave them to the reader.

Remark 2.9. While we already remarked that the above construction is not the only reasonable way to construct $\uparrow_R (d)$, we claim that all such correct constructions are in fact the 'equivalent', in the sense that their equivalence is witnessed by higher cells which are marked as equivalences. The notion of equivalences, or synonymously, the notion of *unitary* cells, will be discussed in the next section.

Definition 2.10. We define left and right transposition for a $(n+1)$-cell f

$$f : \partial_0 f \to \partial_1 f$$

(i) The left transpose, denoted by

$$f^{\uparrow_L} : (\partial_1 f)^* \to (\partial_0 f)^* \; ,$$

is the composite of the diagram

$$\begin{array}{c} (\partial_1 f)^* \\ = (\partial_1 f)^*.\mathrm{id}(\partial_0\partial_0 f) \\ \quad\Big\downarrow {\scriptstyle (\vec{0},|\partial_1 f|,0)\ \downarrow_R(\partial_0 f)} \\ (\partial_1 f)^*.(\partial_0 f).(\partial_0 f)^* \\ \quad\Big\downarrow {\scriptstyle (\vec{0},|\partial_1 f|,|\partial_0 f|)\ f} \\ (\partial_1 f)^*.(\partial_1 f).(\partial_0 f)^* \\ \quad\Big\downarrow {\scriptstyle (\vec{0},0,|\partial_0 f|)\ \uparrow_L(\partial_1 d)} \\ \mathrm{id}(\partial_{|\partial_1 d|}\partial_1 f).(\partial_0 f)^* \\ = (\partial_0 f)^* \end{array}$$

This sequence is indeed a valid diagram as it manifestly satisfies the boundary conditions (1.24).

(ii) The right transpose, denoted by

$$f^{\uparrow_R} : (\partial_1 f)^* \to (\partial_0 f)^* ,$$

is the composite of the diagram

$$\begin{array}{c} (\partial_1 f)^* \\ = \mathrm{id}(\partial_{|\partial_0|}\partial_0 f).(\partial_1 f)^* \\ \quad\Big\downarrow {\scriptstyle (\vec{0},0,|\partial_1 f|)\ \downarrow_L(\partial_0 f)} \\ (\partial_0 f).(\partial_0 f)^*.(\partial_1 f)^* \\ \quad\Big\downarrow {\scriptstyle (\vec{0},|\partial_0 f|,|\partial_1 f|)\ f} \\ (\partial_0 f)^*.(\partial_1 f).(\partial_1 f)^* \\ \quad\Big\downarrow {\scriptstyle (\vec{0},|\partial_0 f|,0)\ \uparrow_R(\partial_1 d)} \\ (\partial_0 f)^*.\mathrm{id}(\partial_0\partial_1 f) \\ = (\partial_0 f)^* \end{array}$$

which again is a valid diagram, as it manifestly satisfies the boundary conditions (1.24) (domain and codomain of shifts are marked in color for better readability).

(iii) Similarly, but under the condition the boundaries of f are duals of diagrams themselves, i.e.

$$\partial_0 f = d_0^*, \partial_1 f = d_1^* ,$$

we can define the following composites in analogy to the definitions above

$$f^{\downarrow_L} := \left\langle\ d_1 \xrightarrow[(\vec{0},|d_1|,0)]{\downarrow_L(d_0)} d_1.d_0^*.d_0 \xrightarrow[(\vec{0},|d_1|,|d_0|)]{f} d_1.d_1^*.d_0 \xrightarrow[(\vec{0},0,|d_0|)]{\uparrow_R(d_1)} d_0\ \right\rangle$$

and

$$f^{\downarrow_R} := \left\langle\ d_1 \xrightarrow[(\vec{0},0,|d_1|)]{\downarrow_R(d_0)} d_0.d_0^*.d_1 \xrightarrow[(\vec{0},|d_0|,|d_1|)]{f} d_0.d_1^*.d_1 \xrightarrow[(\vec{0},|d_0|,0)]{\uparrow_L(d_1)} d_0\ \right\rangle$$

It is important to reiterate that these definitions only apply if the boundaries $\partial_0 f$, $\partial_1 f$ can be written as the duals of some diagrams d_0, d_1 – as a shorthand we will write this as $\partial_0 f, \partial_1 f \in D_n^*$.

Importantly, from the definition of left and right transposition we see that

$$\partial_0 f^{\uparrow_R} = \partial_0 f^{\uparrow_L} = (\partial_1 f)^* \qquad (2.11)$$
$$\partial_1 f^{\uparrow_R} = \partial_1 f^{\uparrow_L} = (\partial_0 f)^*$$

and similarly, if $\partial_0 f = d_0^*, \partial_1 f = d_1^* \in D_n^*$, we have

$$\partial_0 f^{\downarrow_R} = \partial_0 f^{\downarrow_L} = d_1 \qquad (2.12)$$
$$\partial_1 f^{\downarrow_R} = \partial_1 f^{\downarrow_L} = d_0$$

Definition 2.13. Dualities and shifts (and consequently transpositions) generate a broad range of new cells, and it will turn out to be natural to equate some of them. We will do so now, by 'augmenting' Id_{C_n} by certain equivalences. As noted in Remark 1.9, these equations need to be admissible in the sense that all previously defined functions on C_n (namely ∂_0, ∂_1) preserve them.

For dualities we introduce the following

$$\begin{array}{lll} \mathrm{Id}_{C_n} :: & | \quad \langle d \rangle^* = \langle d^* \rangle & , \text{ where } d \in D_n \\ & | \quad p_1 = p_2 & , \text{ where } p_i \in C_n, p_1^* = p_2^* \end{array}$$

The first equation says dualisation and composition commute. The second equation says dualisation is in fact injective. Both equations are seen to be admissible by the definition of boundaries of dualised cells p^* (2.5) and cells from composed diagrams $\langle d \rangle$ (2.3).

We recall from the introduction that the peculiar notation

$$\mathrm{Id}_{C_n} :: \ \dots | \quad a = b \ , \ P(a,b)$$

should indicate a path/equivalence of type $\mathrm{Id}_{C_n}(a,b)$ can be constructed for approriate a, b satisfying $P(a,b)$: Again, *we do not care about the naming of such a path* as C_n is in fact a set.

Example 2.14. For $1(d)$ the previous equalities imply for instance

$$1(d)^* := \langle \mathrm{id}(d) \rangle^* = \langle \mathrm{id}(d)^* \rangle = \langle \mathrm{id}(d) \rangle =: 1(d)$$

i.e. $1(d)^* = 1(d)$ is a fixed point of dualisation. ◊

Next we consider equations related to transposition. We introduce the following

$$\begin{array}{lll} \mathrm{Id}_{C_{n+1}} :: \ \dots & | \quad f^{\uparrow_R} = f^{\uparrow_L} =: f^{\intercal} & , \text{ where } f \in C_{n+1} \\ & | \quad f^{\downarrow_R} = f^{\downarrow_L} =: f^{\perp} & , \text{ where } f \in C_{n+1}, \partial_i f \in D_n^* \\ & | \quad (f^{\intercal})^{\perp} = f & , \text{ where } f \in C_{n+1} \\ & | \quad (f^{\perp})^{\intercal} = f & , \text{ where } f \in C_{n+1}, \partial_i f \in D_n^* \end{array}$$

While the first two equations equate left and right transposition and 'co-transposition', the last two equations record that those two operations are in fact inverse to each other. For the first two equalities admissibility was verified in (2.11) and (2.12). From there it can be equally derived for the last two equations. Further we introduced the shorthand $f^{\intercal}$ to denote the left or right transpose of f, and $f^{\perp}$ to denote its 'co-transpose' if it exists.

Finally, we consider how transposition interacts with special cells we have defined so far, namely identities $1(d) := \langle \mathrm{id}(d) \rangle$ and witnesses of composition $d \Rightarrow \langle d \rangle$. We equate the following

$$\begin{array}{lll} \mathrm{Id}_{C_{n+1}} :: \ \dots & | \quad 1(d)^{\intercal} = 1(d^*) & , \text{ where } d \in D_n \\ & | \quad (d \Rightarrow \langle d \rangle)^{\intercal} = (d^* \Rightarrow \langle d^* \rangle)^* & , \text{ where } d \in D_n \end{array}$$

It can be safely left to the reader to trace back definitions and verify that both equations are admissible (i.e. ∂_0, ∂_1 coincide on the cells that are equated by these two new equivalences). ◊

2.3. **Unitarity.**

Unitarity of a cell marks when this cell witnesses an equivalence between diagrams, and witnesses of composition are one such example. Compositions and unitaries are the 'invertible' cells in some sense. On the other hand, a cell is unitary iff its left and right shift are unitary and not just *mere cells*. This calls for a coinductive definition which we write as follows

Definition 2.15. *Mutual invertibility* for cells $p_1, p_2 \in C_n$, written $\mathrm{inv}(p_1, p_2)$, and *unitarity* of a cell $p \in C_n$, written $\mathrm{is\_uni}(p)$, are defined as follows

$$\begin{aligned} \mathrm{inv}(p_1, p_2) := \exists c_1 &: p_1.p_2 \to \mathrm{id}(\partial_0 p_1) \\ \wedge \exists c_2 &: p_2.p_1 \to \mathrm{id}(\partial_1 p_1) \\ &\text{s.t. } \mathrm{is\_uni}(c_1) \wedge \mathrm{is\_uni}(c_2) \end{aligned}$$

$$\mathrm{is\_uni}(p) := \mathrm{is\_uni}(\uparrow_R(p)) \wedge \mathrm{is\_uni}(\uparrow_L(p))$$

Note that the dimension n is implicit for the predicates inv and is_uni. ♢

Construction 2.16. We postulate unitarity of the following cells:

- Invertibility implies unitarity, that is for $p \in C_n$:
$$(\exists p' \in C_n.\mathrm{inv}(p, p')) \Rightarrow \mathrm{is\_uni}(p)$$
- Identitites are unitary:
$$\mathrm{is\_uni}(1(d))$$
- Witnesses of composition are, as promised, unitary:
$$\mathrm{is\_uni}(d \Rightarrow \langle d \rangle)$$
- Unitarity is preserved under composition, for $d \in D_n$:
$$(\forall i.\mathrm{is\_uni}(p_i d)) \Rightarrow \mathrm{is\_uni}(\langle d \rangle)$$
- Unitarity is preserved under transposition, for $p \in C_n$:
$$\mathrm{is\_uni}(p) \Rightarrow \mathrm{is\_uni}(p^{\intercal})$$

(- By Lemma 2.18, Unitarity is also preserved under dualisation, that is for $p \in C_n$:
$$\mathrm{is\_uni}(p) \Rightarrow \mathrm{is\_uni}(p^*) \quad)$$

♢

Definition 2.17. We define $\mathrm{Uni}_n = \{\, p \mid p \in C_n, \mathrm{is\_uni}(p) \,\}$.

Lemma 2.18. *The following are consequences of the above definition. Let* $p \in C_n$*. Then*

(i) $\mathrm{is\_uni}(p) \Rightarrow \exists p'.\mathrm{inv}(p, p')$
(ii) $\mathrm{is\_uni}(p) \Rightarrow \mathrm{is\_uni}(p^*)$
(iii) *Let* $e : p \to p'$*. If* $\mathrm{is\_uni}(p), \mathrm{is\_uni}(e)$ *then we have* $\mathrm{is\_uni}(p')$.

Proof. Part (i) follows with $p' = p^*$ and setting $c_1 = \uparrow_R(p)$, $c_2 = \uparrow_L(p)$ which indeed satisfies the conditions for invertibility and thus yields

$$\mathrm{is\_uni}(p) \Rightarrow \mathrm{inv}(p, p^*)$$

Part (ii) is a consequence of part (i), invertibility being symmetric $\mathrm{inv}(p_1, p_2) \iff \mathrm{inv}(p_2, p_1)$ and invertibility implying unitarity:

$$\mathrm{is\_uni}(p) \Rightarrow \mathrm{inv}(p, p^*) \Rightarrow \mathrm{inv}(p^*, p) \Rightarrow \mathrm{is\_uni}(p^*)$$

Part (iii) follows from p' being having p^* as a mutual inverse under the stated assumptions: This is witnessed by

$$c_1' := \left\langle\ p'.p^* \xrightarrow[(\vec{0},0,1)]{e^*} p.p^* \xrightarrow{\uparrow_R(p)} \mathrm{id}(\partial_0 p')\ \right\rangle$$

and

$$c_2' := \left\langle \; p^*.p' \xrightarrow[(\vec{0},1,0)]{e^*} p^*.p \xrightarrow{\uparrow_L(p)} \mathrm{id}(\partial_1 p') \; \right\rangle$$

Both c_i' are unitary since they are composites of unitary cells, and thus serve as the required witnesses for

$$\mathrm{inv}(p', p^*)$$

The statement follows from $\mathrm{inv}(p', p^*) \Rightarrow \mathrm{is\_uni}(p')$. □