Type theoretical approaches to opetopes

Cédric Ho Thanh, Pierre-Louis Curien, Samuel Mimram

▶ **To cite this version:**

Cédric Ho Thanh, Pierre-Louis Curien, Samuel Mimram. Type theoretical approaches to opetopes. Higher Structures, 2022, 6 (1), pp.80 - 181. ⟨10.21136/hs.2022.02⟩. ⟨hal-04244406⟩

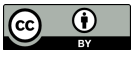
HAL Id: hal-04244406

https://hal.science/hal-04244406v1

Submitted on 16 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Higher Structures 6(1):80–181, 2022.

HIGHER STRUCTURES

Type theoretical approaches to opetopes

Pierre-Louis Curien[a], Cédric Ho Thanh[a] and Samuel Mimram[b]

[a]*Research Institute for Foundations of Computer Science (IRIF), Paris University and CNRS, France*
[b]*Computer Science Laboratory of École Polytechnique (LIX), Palaiseau, France*

Abstract

Opetopes are algebraic descriptions of shapes corresponding to compositions in higher dimensions. As such, they offer an approach to higher-dimensional algebraic structures, and in particular, to the definition of weak ω-categories, which was the original motivation for their introduction by Baez and Dolan. They are classically defined inductively (as free operads in Leinster's approach, or as zoom complexes in the formalism of Kock et al.), using abstract constructions making them difficult to manipulate with a computer.

In this paper, we present two purely syntactic descriptions of opetopes as sequent calculi, the first using variables to implement the compositional nature of opetopes, the second using a calculus of higher addresses. We prove that well-typed sequents in both systems are in bijection with opetopes as defined in the more traditional approaches. Additionally, we propose three variants to describe opetopic sets. We expect that the resulting structures can serve as natural foundations for mechanized tools based on opetopes.

Communicated by: Richard Garner.
Received: 26th April, 2019. Accepted: 12th April, 2022.
MSC: Primary 18D50; Secondary 03B15.
Keywords: Opetope, Opetopic set, Type theory, Polynomial functor.

1. Introduction

1.1 Opetopes Opetopes were originally introduced by Baez and Dolan in order to formulate a definition of weak ω-categories [1]. Their name reflects the fact that they encode the possible shapes for higher-dimensional operations: they are *ope*ration poly*topes*. Over the recent years, they have been the subject of many efforts to provide a good definition that would allow exploring their combinatorics [3, 10, 18]. One of the most commonly used nowadays is the formulation based

Email addresses: curien@irif.fr (P.L. Curien)
cedric.hothanh@irif.fr (C. Ho Thanh)
samuel.mimram@lix.polytechnique.fr (S. Mimram)

© P.L. Curien, C. Ho Thanh and S. Mimram, 2022, under a Creative Commons Attribution 4.0 International License.

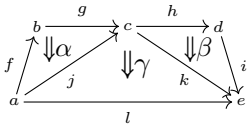
DOI: 10.21136/HS.2022.02

on polynomial functors and the corresponding graphical representation using "zoom complexes" [16].

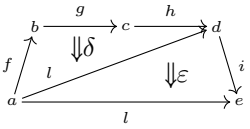
In order to grasp quickly the nature of opetopes, consider a sequence of four composable arrows

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d \xrightarrow{i} e$$

There are various ways we can compose them. For instance, we can compose f with g, as well as h with i, and then $g \circ f$ with $i \circ h$. Or we can compose f, g and h together all at once, and then the result with i. These two diagrams for composing can respectively be pictured



and

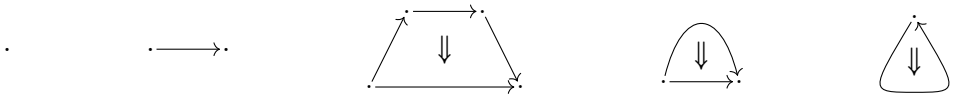


(1.1)

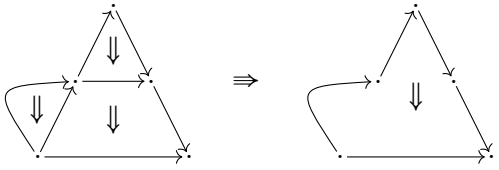
From there, the general idea of getting "higher-dimensional" is that we should take these compositions as "2-operations", which can be composed in various ways. For instance, in the first case, we can compose α with γ, and then β with the result, or all three at once, and so on. The opetopes describe all the ways in which these compositions can be meaningfully specified, in arbitrary dimension. We are also interested in *opetopic sets*, which are the carriers of such composition operations. It is in this setting that Baez and Dolan proposed a definition of weak ω-category in [1].

We can expect (and it is indeed the case) that the combinatorics of these objects is not easy to describe. In particular, a representation which is adapted to computer manipulations and proofs is desirable: we can for instance mention the *Opetopic* proof assistant for higher categories [4], which is based on opetopes. Recently, Mimram and Finster have introduced a language of a type-theoretical flavour to describe globular weak ω-categories [5]. Our goal is to achieve a similar type-theoretic presentation for opetopic weak ω-categories [1, 11]. Here, type-theoretical is meant in the very broad sense of typed languages: there is a syntax for raw expressions, that is then filtered by a type system inferring valid judgments, where the judgments have the form of a claim that a given term indeed represents an opetope (or an opetopic set) in a certain context.

Let us now informally define opetopes. At the basis of the architecture, there is a unique 0-opetope (i.e. opetope of dimension 0), drawn as a point. An $(n+1)$-opetope (i.e. an opetope of dimension $n+1$) is made out of n-opetopes, and has a source and a target. In small dimensions, opetopes can be described using drawings of the kind above. For instance, the following are, from left to right, the unique 0-opetope, the unique 1-opetope, drawn as an arrow, and three 2-opetopes, respectively,



while the following drawing represents a 3-opetope:



In these drawings, the target of the 1-opetope is a point, the target of the 2-opetopes is the arrow at the bottom, and the target of the 3-opetope is the 2-opetope on the right, while the source of

the 1-opetope is again a point, the source of the 2-opetopes is a diagram made of arrows, and the source of the 3-opetope is the diagram made of 2-opetopes on the left of the central arrow.

At this stage, we can already make the following observations:

(1) each drawing contains only one top-dimensional cell, whose dimension determines the dimension of the opetope,

(2) if an opetope has a non empty source, then its source and target have the same source (which might be empty) and the same target,

(3) if an opetope has an empty source, then its target has the same source and target,

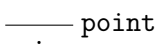
(4) an $(n+1)$-cell can have multiple n-cells in its source (including none) but always has exactly one n-cell in its target.

By the above remarks, an n-opetope has a unique top-dimensional cell α, whose target consists of one cell β whose source and target are the same as the ones of the source of α. It follows recursively that the opetope is entirely determined by the source of α, which we call the associated *pasting diagram*, which is of dimension $n-1$. The opetope associated to a pasting diagram of dimension n can be obtained by adding a single cell β of dimension n parallel to the pasting diagram, and an $(n+1)$-cell α from the pasting diagram to β.

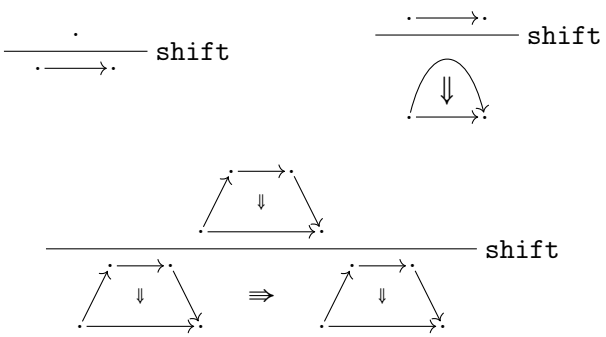
This article aims to provide type-theoretical tools making opetopes easier to manipulate than their classical definitions. The *unnamed* (or *anonymous*) approaches of section 4 are purely relying on a calculus of higher addresses, and on a syntactic notion of *preopetope*. Opetopes are then well-formed preopetopes according to the derivation system Opt$^?$ presented in definition 4.17, while opetopic sets are derivable contexts in system OptSet$^?$. The *named* approaches of section 3, while slightly more complicated, leverage the idea of cell naming to produce user friendlier tools and results. As such it comes in two variants: Opt$^!$ for describing opetopes, and OptSet$^!$ for opetopic sets, introduced in definitions 3.23 and 3.71 respectively. The Python implementation of [13] is also discussed.

1.2 Generating opetopes Our two opetope derivation systems Opt$^!$ (definition 3.23) and Opt$^?$ (definition 4.17) are based on the observation that opetopes are precisely all the shapes one can generate with the following operations.

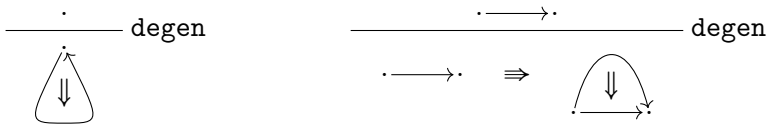
(1) *Introduction of a point.* There is a unique 0-opetope (the *point*).



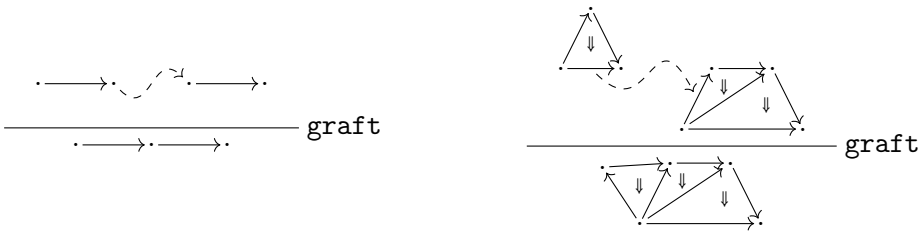
(2) *Shift to the next dimension.* Given an n-opetope ω, we can form the $(n+1)$-globe whose source and target are ω, as illustrated below. It can geometrically be thought of as the "extrusion" of ω.



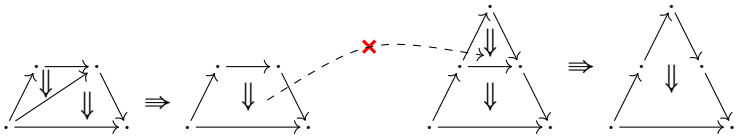
(3) *Introduction of degeneracies.* Given an n-opetope ω, we can build an $(n+2)$-opetope with empty source, whose target is the globe at ω, as illustrated below for $n = 0$ and $n = 1$:



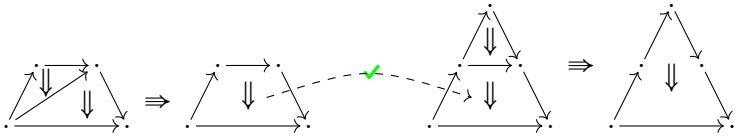
(4) *Grafting.* Given an $(n+1)$-opetope α and an $(n+1)$-pasting diagram β such that the source of β contains an n-cell of the same shape as the target of α, we can *graft* α to β:



Ill-formed graftings may occur with n-pasting diagrams, for $n \geq 3$, and the side condition is necessary to rule them out. Here is an example the `graft` rule will not allow: we deal with a 3-pasting diagram on the right of the dashed arrow (that comprises a unique 3-opetope), and the dashed arrow indicates that we attempt to graft the 3-opetope on the left (whose target shape is a trapezoid) onto the triangle shaped cell on the right

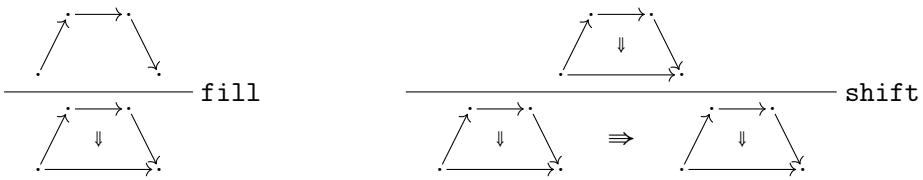


However, grafting onto the lower trapezoid of the right opetope is possible:



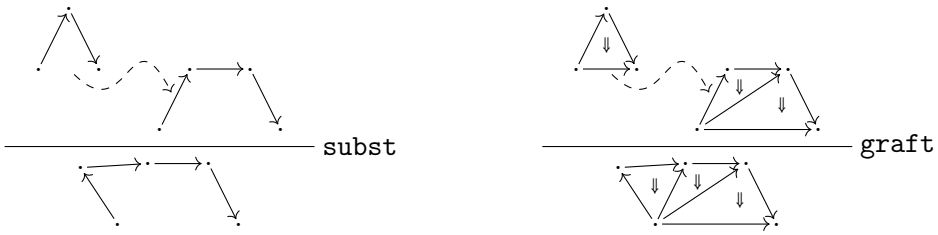
As previously mentioned, an opetope is completely determined by its source pasting diagram, i.e. "arrangement" of source faces (the dichotomy between opetopes and pasting diagrams is more thoroughly discussed in section 2.2.2). We can reformulate rules `shift` and `graft` with this point of view to respectively obtain:

(1) *Filling of pasting diagrams.* Given an n-pasting diagram, we may "fill" it by adding a target n-cell, and a top dimensional $(n+1)$-cell. We illustrate an instance of this rule on the left, and invite the reader to compare it with the instance of `shift` on the right:

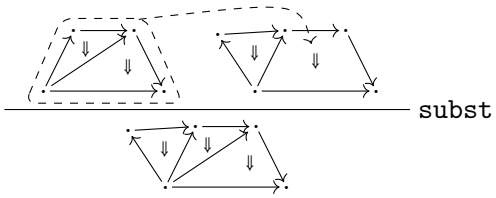


(2) *Substitution*, which consists in replacing a cell in a pasting diagram by another "parallel" pasting diagram. As before, we illustrate an instance if this rule on the left, and invite

the reader to compare it with the instance of `graft` on the right:



Substitution can be further illustrated as follows:



Rules `point` and `degen` can also be reformulated in term of pasting diagrams, but representing them graphically is not as insightful.

1.3 Plan In section 2, we expound some prerequisites and general ideas motivating our syntactic approaches. We then present the "unnamed" and "named" systems for opetopes and opetopic sets in sections 3 and 4 respectively. Those two chapters can be read independently, but have a similar structure:

(1) firstly, we present the syntactic constructs and inference rules of systems Opt$^!$ (named) and Opt$^?$ (unnamed) for opetopes, in sections 3.1 and 4.1 respectively;
(2) we then prove the equivalence with opetopes as defined in [16] (which we shall refer to as *polynomial opetopes*) in sections 3.2 and 4.2;
(3) we showcase the Opt$^!$ and Opt$^?$ systems in examples in sections 3.3 and 4.3;
(4) we discuss the Python implementation of Opt$^!$ and Opt$^?$ of [13] in sections 3.4 and 4.5;
(5) in the second part of those chapters, we present the syntactic constructs and inference rules of systems OptSet$^!$ (named) and OptSet$^?$ (unnamed) for opetopic sets, in sections 3.5 and 4.6 respectively;
(6) we then prove the equivalence with opetopic sets of [12] in sections 3.6 and 4.7;
(7) we showcase the OptSet$^!$ and OptSet$^?$ systems in examples in sections 3.7 and 4.8;
(8) we discuss the Python implementation of OptSet$^!$ and OptSet$^?$ sections 3.8 and 4.9;
(9) for the named approach of section 3, we present the additional "mixed" system OptSet$^!_{\mathrm{M}}$ in section 3.9, with the same plan as previously presented.

The equivalence proofs with polynomial opetopes of sections 3.2 and 4.2 rely on a precise definition of the Baez–Dolan $(-)^+$ construction [16, 1], which is presented in appendix A.

1.4 Related works Our syntactic opetopes are shown in sections 3.2 and 4.2 to be equivalent to the polynomial opetopes (or "zoom complexes") of Kock et al. [16], which are themselves equivalent to Leinster's opetopes [18]. It is known that the latter are incompatible with Cheng's opetopes [3], which should be thought of as being symmetric. There is a closely related notion of *multitope* [11, 9] which is defined in terms of multicategories (whence the *multi*) instead of operads (*ope*); the two notions can be shown to be equivalent [12]. A syntax for multitopes was

proposed in [11], where however not all the desired computations have been given algorithmic formulations.

The *Opetopic* proof assistant [4] for weak higher categories relies on the notion of higher-dimensional tree. In that system, the notion of opetope is built-in, so that we have to trust the implementation. In contrast, the present approach allows us to reason about the construction of opetopes. We moreover believe that the ability to reason by induction on the proof trees, together with the very explicit nature of our syntaxes, will allow for optimizations in the automated manipulations of opetopes.

Another proof assistant for weak higher categories, called *CaTT* [5], starts from the same idea of generating well-formed pasting diagrams through inference rules. However, it is based on globular shapes instead of opetopic ones, making a comparison with the present work difficult: since their introduction, people have unsuccessfully tried to compare the resulting respective categorical formalisms; we hope that their formulation in a common logical language might be of help in this task.

We should also mention here the *Globular* proof assistant [2], also based on globular shapes, which is quite popular, notably thanks to its nice graphical interface.

Acknowledgments We would like to thank the anonymous referee for their scrupulous review and valuable advices. The second author has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement №665850.

2. Preliminaries

This chapter introduces the theory of polynomial functors, trees, and monads of [15, 7], and applies it to the definition of opetopes in the style of Kock et. al. [16].

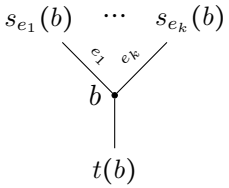
2.1 Polynomial endofunctors and trees

2.1.1 Definitions

Definition 2.1 (Polynomial endofunctor [7, paragraph 1.4])**.** A *polynomial endofunctor* P is a diagram in $\mathcal{S}\mathrm{et}$ of the form

$$I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} I. \tag{2.2}$$

We use the following terminology for a polynomial endofunctor P as in equation (2.2), which is motivated by the intuition that a polynomial endofunctor encodes a multi-sorted signature of function symbols. The elements of B are called the *nodes* or *operations* of P, and for every node b, the elements of the fibre $E(b) := p^{-1}(b)$ are called the *inputs* of b. The elements of I are called the *colors* or *sorts* of P. For every input e of a node b, we denote its color by $s_e(b) := s(e)$.



In this paper, all polynomial functors are assumed to be *finitary*, meaning that all the fibers of p in (2.2) are finite. In other words, every operation has finitely many inputs.

Definition 2.3 (Morphism of polynomial endofunctor)**.** A morphism f from a polynomial endofunctor P over I (on the first row) to a polynomial endofunctor P' over I' (on the second row) is a commutative diagram of the form

$$\begin{array}{ccccccc} I & \xleftarrow{s} & E & \xrightarrow{p} & B & \xrightarrow{t} & I \\ {\scriptstyle f_0}\downarrow & & {\scriptstyle f_2}\downarrow & \lrcorner & {\scriptstyle f_1}\downarrow & & {\scriptstyle f_0}\downarrow \\ I' & \xleftarrow{s'} & E' & \xrightarrow{p'} & B' & \xrightarrow{t'} & I' \end{array}$$

where the middle square is cartesian (i.e. is a pullback square). If P and P' are both polynomial endofunctors over I, then a morphism from P to P' *over* I is a commutative diagram as above, but where f_0 is required to be the identity [15, paragraph 0.1.3] [16, section 2.5]. Let $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}$ denote the category of polynomial endofunctors and morphisms of polynomial endofunctors, and $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$ the category of polynomial functors over I and morphisms of polynomial endofunctors over I.

Remark 2.4. A polynomial functor P as in equation (2.2) induces a functor $P = t_! p_* s^* : \mathcal{S}\mathrm{et}/I \longrightarrow \mathcal{S}\mathrm{et}/I$, where $t_!$ is the dependent sum along t, p_* is the dependent product along p, and s^* is the pullback along s. Furthermore, morphisms between polynomial functors induce cartesian natural transformations[1] [7, paragraph 2.1].

Definition 2.5 (Polynomial tree [15, section 1.0.3])**.** A polynomial endofunctor T given by

$$T_0 \xleftarrow{s} T_2 \xrightarrow{p} T_1 \xrightarrow{t} T_0$$

is a *polynomial tree* (or just *tree*) if

(1) the sets T_0, T_1 and T_2 are finite (in particular, each node has finitely many inputs); by convention we assume $T_0 \neq \varnothing$;
(2) the map t is injective;
(3) the map s is injective, and the complement of its image $T_0 - \operatorname{im} s$ has a single element, called the *root*;
(4) let $T_0 = T_2 + \{r\}$, with r the root, and define the *walk-to-root* function σ by $\sigma(r) = r$, and otherwise $\sigma(e) = tp(e)$; then we ask that for all $x \in T_0$, there exists $k \in \mathbb{N}$ such that $\sigma^k(x) = r$.

We call the colors of a tree its *edges* and the inputs of a node the *input edges* of that node.

Let $\mathcal{T}\mathrm{ree}$ be the full subcategory of $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}$ whose objects are trees. Note that it is the category of *symmetric* or *non-planar* trees (the automorphism group of a tree is in general non-trivial) and that its morphisms correspond to inclusions of non-planar subtrees. An *elementary tree* is a tree with at most one node, and we write $\mathcal{T}\mathrm{ree}_{\mathrm{elem}}$ for the full subcategory of $\mathcal{T}\mathrm{ree}$ spanned by elementary trees.

Definition 2.6 (P-tree)**.** For $P \in \mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}$, the category $\mathrm{tr}\, P$ of *P-trees* is a chosen skeleton of the slice category $\mathcal{T}\mathrm{ree}/P$.

[1] Recall that a natural transformation is *cartesian* if all its naturality squares are pullback squares.

Notation 2.7. A P-tree $T \in \mathrm{tr}\, P$ is a morphism from a polynomial tree, which we shall denote by $\langle T \rangle$, to P, as in $T : \langle T \rangle \longrightarrow P$. We point out that $\langle T \rangle_1$ is the set of nodes of the P-tree T, while $T_1 : \langle T \rangle_1 \longrightarrow P_1$ provides a *decoration* of the nodes of $\langle T \rangle$ by operations of P, and likewise for edges.

Definition 2.8 (Address)**.** Let $T : \langle T \rangle \longrightarrow P$ be a P-tree, and σ be its walk-to-root function (definition 2.5). We define the *address* function $\&$ on edges of $\langle T \rangle$ inductively as follows:

(1) if r is the root edge of $\langle T \rangle$, let $\& r := []$;
(2) if $i \in \langle T \rangle_0 - \{r\}$ and if $\& \sigma(i) = [x]$, where σ is the walk-to-root function of $\langle T \rangle$, define $\& i := [x\ T_2(e)]$, where $e \in \langle T \rangle_2$ is the unique element such that $s(e) = i$.

Thus an address is a sequence of elements of P_2, *which we always enclose with brackets*. The address of a node $b \in \langle T \rangle_1$ is simply $\& b := \& t(b)$. Note that this function is injective since t is. Let $T^\bullet$ denote its image, the set of *node addresses* of T, and let $T^!$ be the set of addresses of leaf edges, i.e. those edges not in the image of t.

If $b \in \langle T \rangle_1$, let $\& b := \& t(b)$. Furthermore, $\&$ is still injective since the square

$$\begin{array}{ccc} \langle T \rangle_2 & \xrightarrow{p} & \langle T \rangle_1 \\ {\scriptstyle T_2}\downarrow & & \downarrow{\scriptstyle T_1} \\ P_2 & \xrightarrow{p} & P_1 \end{array}$$

is cartesian. If $b \in \langle T \rangle_1$ has address $\& b = [p]$, write $\mathsf{s}_{[p]} T := T_1(b)$ for the decoration of node b. Likewise, if an edge i has address $[q]$, let $\mathsf{e}_{[q]} T := T_0(i)$ be its decoration.

Definition 2.9 (Elementary P-trees)**.** Let P be a polynomial endofunctor as in equation equation (2.2). For $i \in I$, define $\mathsf{I}_i \in \mathrm{tr}\, P$ as having underlying tree

$$\{i\} \longleftarrow \varnothing \longrightarrow \varnothing \longrightarrow \{i\}, \tag{2.10}$$

along with the obvious morphism to P, that which maps i to $i \in I$. This corresponds to a tree with no nodes and a unique edge, decorated by i. For $b \in B$, define $\mathsf{Y}_b \in \mathrm{tr}\, P$, the *corolla* at b, as having underlying tree

$$s(E(b)) + \{*\} \xleftarrow{s} E(b) \longrightarrow \{b\} \longrightarrow s(E(b)) + \{*\}, \tag{2.11}$$

where the right map sends b to $*$, and where the morphism $\mathsf{Y}_b \longrightarrow P$ is the identity on $s(E(b)) \subseteq I$, maps $*$ to $t(b) \in I$, is the identity on $E(b) \subseteq E$, and maps b to $b \in B$. This corresponds to a P-tree with a unique node, decorated by b. Observe that for $T \in \mathrm{tr}\, P$, giving a morphism $\mathsf{I}_i \longrightarrow T$ is equivalent to specifying the address $[p]$ of an edge of T decorated by i. Likewise, morphisms of the form $\mathsf{Y}_b \longrightarrow T$ are in bijection with addresses of nodes of T decorated by b.

Remark 2.12. Let P be a polynomial endofunctor as in equation (2.2).

(1) Let $i \in I$ be a color of P. Since I_i does not have any nodes, the set $\mathsf{I}_i^\bullet$ of its node addresses is empty. On the other hand, the set of its leaf addresses is $\mathsf{I}_i^! = \{[]\}$, since the unique leaf is the root edge.
(2) Let $b \in B$ be an operation of P. Then $\mathsf{Y}_b^\bullet = \{[]\}$ since the only node is that above the root edge. For leaves, we have $\mathsf{Y}_b^! = \{[e] \mid e \in E(b)\}$.

Definition 2.13 (Grafting)**.** For $S, T \in \operatorname{tr} P$, $[l] \in S^!$ such that the leaf of S at $[l]$ and the root edge of T are decorated by the same $i \in I$, define the *grafting* $S \circ_{[l]} T$ of S and T on $[l]$ by the following pushout (in $\operatorname{tr} P$):

$$\begin{array}{ccc} \mathsf{I}_i & \xrightarrow{[]} & T \\ {\scriptstyle [l]}\downarrow & \ulcorner & \downarrow \\ S & \longrightarrow & S \circ_{[l]} T. \end{array} \tag{2.14}$$

In particular,

$$\begin{aligned} (S \underset{[l]}{\circ} T)^\bullet &= S^\bullet + \{[lp] \mid [p] \in T^\bullet\}, \\ (S \underset{[l]}{\circ} T)^! &= S^! - \{[l]\} + \{[lp] \mid [p] \in T^!\}. \end{aligned}$$

Note that if S (resp. T) is a trivial tree, then $S \circ_{[l]} T = T$ (resp. S). We assume, by convention, that the grafting operator $\circ$ associates to the right. In particular $\mathsf{I}_i \circ_{[]} T = T$ and $S \circ_{[l]} \mathsf{I}_i = S$.

Notation 2.15 (Total grafting). Let $T \in \operatorname{tr} P$, write $T^! = \{[l_1], \dots, [l_k]\}$, take $U_1, \dots, U_k \in \operatorname{tr} P$, and assume that the grafting $T \circ_{[l_i]} U_i$ is defined for all i. Then the *total grafting* will be denoted concisely by

$$T \bigcirc_{[l_i]} U_i = (\cdots (T \underset{[l_1]}{\circ} U_1) \underset{[l_2]}{\circ} U_2 \cdots) \underset{[l_k]}{\circ} U_k. \tag{2.16}$$

It is easy to see that the result does not depend on the order in which the graftings are performed.

Proposition 2.17 (Induction principle for trees [15, proposition 1.1.21])**.** *Let P be a polynomial endofunctor as in equation (2.2). A P-tree is either of the form I_i for some $i \in I$, or can be uniquely decomposed as*

$$\mathsf{Y}_b \bigcirc_{[e]} T_e,$$

where $b \in B$, e ranges over $E(b)$, and $T_e \in \operatorname{tr} P$.

Remark 2.18. The induction principle of proposition 2.17 describes opetopes as something atomic (the root node) onto which more opetopes have been attached (i.e. grafted). However, in most proofs, as well as in the inference rules for our syntactical approaches, we rely on the a "dual" induction principle: a P-tree is either of the form I_i for some $i \in I$, or can be decomposed as $T \circ_{[l]} \mathsf{Y}_b$, where $T \in \operatorname{tr} P$, $[l] \in T^!$, and $b \in B$. In the second case, we thus grafted atoms onto trees. Note that this time, the decomposition is not unique.

Definition 2.19 (Substitution)**.** Let T be a P-tree, $[p] \in T^\bullet$, and $b = \mathsf{s}_{[p]} T$. Then T can be decomposed so as to isolate the node of T at address $[p]$:

$$T = A \underset{[p]}{\circ} \mathsf{Y}_b \bigcirc_{[e_i]} B_i, \tag{2.20}$$

where $E(b) = \{e_1, \dots, e_k\}$, and $A, B_1, \dots, B_k \in \operatorname{tr} P$. For U a P-tree with a bijection $\wp : U^! \longrightarrow E(b)$ over I, and such that the decoration of the root edge of U equals $t(b)$, we define the *substitution* $T \,\square_{[p]}\, U$ (leaving $\wp$ implicit) as

$$T \underset{[p]}{\square} U := A \underset{[p]}{\circ} U \bigcirc_{\wp^{-1} e_i} B_i. \tag{2.21}$$

In other words, the node at address $[p]$ in T has been replaced by U, and the map $\wp$ provided "rewiring instructions" to connect the leaves of U to the rest of T.

2.1.2 The polynomial Baez–Dolan $(-)^+$ construction We now give a brief definition of the $(-)^+$ construction, see appendix A for details.

Remark 2.22. [7, section 1.11] The composite of two polynomial functors is polynomial. In details, if P and P' are as in

$$I \xleftarrow{s} E \longrightarrow B \xrightarrow{t} I, \qquad I \xleftarrow{u} F \longrightarrow C \xrightarrow{v} I,$$

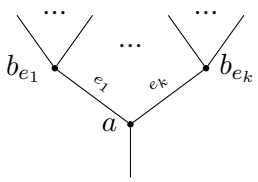
then the composite functor $P'P : \mathcal{S}\mathrm{et}/I \longrightarrow \mathcal{S}\mathrm{et}/K$ is also polynomial. Its underlying diagram is given by

$$I \xleftarrow{x} G \longrightarrow D \xrightarrow{y} I,$$

where

$$D := \{(a, (b_e \mid e \in F(a))) \mid a \in C, b_e \in B, t(b_e) = u_e(a)\},$$
$$G\,(a, (b_e \mid e)) := \sum_{e \in F(a)} E(b_e),$$

where y maps $(a, (b_e \mid e)) \in D$ to $v(a) \in K$, and if $f \in E(b_e)$, then x_f maps $(a, (b_e \mid e))$ to $s_f(b_e)$. Intuitively, D is just the set of trees of uniform height 2 (a.k.a. trees with two levels), where the root node is decorated by an element C, while the nodes on the second level are decorated by elements of B:



here, $a \in C$, $F(a) = \{e_1, \ldots, e_k\}$, and $b_{e_i} \in B$ is such that $t(b_{e_i}) = u(e_i)$. The set of inputs of such a tree is $\sum_i E(b_{e_i})$.

This construction gives $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$, the structure of a monoidal category. Together with remark 2.4 we have a fully faithful functor

$$\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I) \longrightarrow \mathcal{C}\mathrm{art}(\mathcal{S}\mathrm{et}/I),$$

the latter being the category of endofunctors of $\mathcal{S}\mathrm{et}/I$ and cartesian natural transformations[2].

Definition 2.23 (Polynomial monad, classical definition)**.** A *polynomial monad over* I is a monoid object in $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$. Equivalently, it is a polynomial functor endowed with the structure of a cartesian monad[3]. Let $\mathcal{P}\mathrm{oly}\mathcal{M}\mathrm{nd}(I)$ be the category of polynomial monads over I and morphisms of polynomial endofunctors over I that are also monad morphisms.

Definition 2.24 ($(-)^\star$ construction)**.** Given a polynomial endofunctor P as in equation (2.2), we define a new polynomial endofunctor $P^\star$ as

$$I \xleftarrow{s} \mathrm{tr}^{\shortmid}\, P \xrightarrow{p} \mathrm{tr}\, P \xrightarrow{t} I \tag{2.25}$$

where $\mathrm{tr}^{\shortmid}\, P$ is the set of P-trees with a marked (or distinguished) leaf, where s maps a P-tree with a marked leaf to the decoration of that leaf, p forgets the marking, and t maps a P-tree to

[2]We recall that a natural transformation is *cartesian* if all its naturality squares are cartesian.
[3]Recall that a *cartesian monad* is a monad (T, η, μ) such that T preserves pullbacks, and such that the natural transfornations η and μ are cartesian.

the decoration of its root. Remark that for $T \in \operatorname{tr} P$ we have $p^{-1}(T) \cong T^!$, and in particular, $P^\star$ is finitary.

Clearly, there is an inclusion $\eta_P : P \longrightarrow P^\star$, mapping $b \in B$ to $\mathsf{Y}_b \in \operatorname{tr} P$, and $e \in E(b)$ to $[e] \in \mathsf{Y}_b^!$ (see remark 2.12). There is also a morphism $\mu_P : P^{\star\star} \longrightarrow P^\star$ defined inductively as follows:

$$\begin{array}{lll} (\mu_P)_1 : \operatorname{tr} P^\star \longrightarrow \operatorname{tr} P & & \\ \qquad\qquad \mathsf{I}_i \longmapsto \mathsf{I}_i & & i \in I \\ \mathsf{Y}_T \bigcirc\limits_{\substack{[[l]] \\ [l]\in T^!}} X_{[l]} \longmapsto T \bigcirc\limits_{[l]\in T^!} (\mu_P)_1(X_{[l]}) & & T \in \operatorname{tr} P, X_{[l]} \in \operatorname{tr} P^\star. \end{array}$$

Informally, it maps a tree of P-tree Y to the P-trees $(\mu_P)_1(Y)$ obtained by grafting all the P-trees decorating the nodes of Y together.

Finally, the morphisms η_P and μ_P are natural in P, and give $(-)^\star$ the structure of a monad on $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$.

Theorem 2.26. *The polynomial functor $P^\star$ has a canonical structure of polynomial monad. Further, the functor $(-)^\star$ is left adjoint to the forgetful functor $\mathcal{P}\mathrm{oly}\mathcal{M}\mathrm{nd}(I) \longrightarrow \mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$, and the adjunction is monadic.*

Proof. The first two claims are proved in [15, proposition 1.2.8], and monadicity is proved in appendix A.1. See also [8, corollary 5.1.5] where monadicity it is proved in the more general ∞-setting. □

Definition 2.27 (Target and readdressing map)**.** Let M be a polynomial functor as in

$$I \xleftarrow{\ s\ } E \xrightarrow{\ p\ } B \xrightarrow{\ t\ } I.$$

Assume further that M is a $(-)^\star$-algebra, and write its structure map $M^\star \longrightarrow M$ as

$$\begin{array}{ccccccc} I & \longleftarrow & \operatorname{tr}^! M & \longrightarrow & \operatorname{tr} M & \longrightarrow & I \\ \| & & \wp\downarrow & \lrcorner & \downarrow\mathsf{t} & & \| \\ I & \xleftarrow{\ s\ } & E & \xrightarrow{\ p\ } & B & \xrightarrow{\ t\ } & I. \end{array} \tag{2.28}$$

For $T \in \operatorname{tr} M$, we call $\wp_T : T^! \xrightarrow{\cong} E(\mathsf{t}T)$ the *readdressing* function of T, and $\mathsf{t}T \in B$ is called the *target* of T. If we think of an element $b \in B$ as the corolla Y_b, then the target map t "contracts" a tree to a corolla, and since the middle square is a pullback, the number of leaves is preserved. The map $\wp_T$ establishes a coherent correspondence between the set $T^!$ of leaf addresses of a tree T and the elements of $E(\mathsf{t}T)$.

Definition 2.29 (Baez–Dolan $(-)^+$ construction)**.** Let M be a polynomial monad as in equation (2.2), and define its *Baez–Dolan construction* M^+ to be

$$B \xleftarrow{\ \mathsf{s}\ } \operatorname{tr}^\bullet M \xrightarrow{\ p\ } \operatorname{tr} M \xrightarrow{\ \mathsf{t}\ } B \tag{2.30}$$

where $\operatorname{tr}^\bullet M$ is the set of M-trees with a marked (or distinguished) node, where s maps an M-tree with a marked node to the label of that node, p forgets the marking, and t is the target map of definition 2.27. If $T \in \operatorname{tr} M$, remark that $p^{-1}T = T^\bullet$ is the set of node addresses of T, and in particular, P^+ is finitary. If $[p] \in T^\bullet$, then $\mathsf{s}[p] := \mathsf{s}_{[p]}\, T$.

Remark 2.31 (Nested addresses). Let M be a polynomial monad, and $T \in \mathrm{tr}\, M^+$. Then the nodes of T are decorated in M-trees, and its edges by operations of M. Assume that $U \in \mathrm{tr}\, M$ decorates some node of T, say $U = \mathsf{s}_{[p]}\, T$ for some node address $[p] \in T^\bullet$.

(1) The input edges of that node are in bijection with $U^\bullet$. In particular, the address of those input edges are of the form $[p[q]]$, where $[q]$ ranges over $U^\bullet$. This really motivates enclosing addresses in brackets.
(2) On the other hand, the output edge of that node is decorated by $\mathsf{t}U$ (where t is defined in definition 2.27).

Theorem 2.32. *The polynomial endofunctor M^+ has a canonical structure of a polynomial monad.*

Proof. The full construction of this structure is given in appendix A.2. □

2.2 Opetopes

2.2.1 Definition

Definition 2.33 (The $\mathfrak{Z}^n$ monad)**.** Let $\mathfrak{Z}^0$ be the identity polynomial monad on $\mathcal{S}\mathrm{et}$, which has one color and one operation with one input. We write it as

$$\{\blacklozenge\} \longleftarrow \{*\} \longrightarrow \{\blacksquare\} \longrightarrow \{\blacklozenge\}.$$

For $n \geq 1$, let $\mathfrak{Z}^n := (\mathfrak{Z}^{n-1})^+$, and write it as $\mathfrak{Z}^n$ as

$$\mathbb{O}_n \xleftarrow{\mathsf{s}} E_{n+1} \xrightarrow{p} \mathbb{O}_{n+1} \xrightarrow{\mathsf{t}} \mathbb{O}_n, \tag{2.34}$$

i.e. for all $n \in \mathbb{N}$, $\mathbb{O}_n$ is the set of colors of $\mathfrak{Z}^n$.

Definition 2.35 (Opetope)**.** An *n-dimensional opetope* (or *n-opetope* for short) ω is simply an element of $\mathbb{O}_n$, and we write $\dim \omega = n$. If $n \geq 2$, then n-opetopes are exactly the $\mathfrak{Z}^{n-2}$-trees. In this case, an opetope $\omega \in \mathbb{O}_n$ is called *degenerate* if its underlying tree has no nodes (and thus consists of a unique edge), so that $\omega = \mathsf{I}_\phi$ for some $\phi \in \mathbb{O}_{n-2}$. We say that ω it is an *endotope* it its underlying tree has exactly one node, i.e. $\omega = \mathsf{Y}_\psi$ for some $\psi \in \mathbb{O}_{n-1}$.

Following equation (2.28), for $n \geq 2$ and $\omega \in \mathbb{O}_n$, the structure of polynomial monad $(\mathfrak{Z}^{n-2})^\star \longrightarrow \mathfrak{Z}^{n-2}$ gives a bijection $\wp_\omega : \omega^! \longrightarrow (\mathsf{t}\omega)^\bullet$ between the leaves of ω and the nodes of $\mathsf{t}\omega$, preserving the decoration by $(n-2)$-opetopes.

Example 2.36. (1) The unique 0-opetope is denoted $\blacklozenge$ and called the *point*.
(2) The unique 1-opetope is denoted $\blacksquare$ and called the *arrow*.
(3) If $n \geq 2$, then $\omega \in \mathbb{O}_n$ is a $\mathfrak{Z}^{n-2}$-tree, i.e. a tree whose nodes are labeled in $(n-1)$-opetopes, and edges are labeled in $(n-2)$-opetopes. In particular, 2-opetopes are $\mathfrak{Z}^0$-trees, i.e. linear trees, and thus in bijection with $\mathbb{N}$. We will refer to them as *opetopic integers*, and write $\mathbf{n}$ for the 2-opetope having exactly n nodes.
(4) A 3-opetope is a $\mathfrak{Z}^1$-tree, i.e. a planar tree.
(5) A 4-opetope is a $\mathfrak{Z}^2$-tree. Unfolding definitions, if $\omega : \langle\omega\rangle \longrightarrow \mathfrak{Z}^2$, then nodes of ω are decorated by elements of $\mathbb{O}_3$, i.e. planar trees. Further, if $x \in \langle\omega\rangle_1$ is a node of ω, then ω_2 exhibits a bijection between the input edges of x and the nodes of $\omega_1(x) \in \mathbb{O}_3$.

Proposition 2.37. *Let $\omega \in \mathbb{O}_n$ with $n \geq 2$. We have the following.*

(1) If ω is degenerate, say $\omega = \mathsf{I}_\phi$ for some $\phi \in \mathbb{O}_{n-2}$, then $\mathtt{t}\omega = \mathsf{Y}_\phi$, and $\wp_\omega : \omega^! = \{[]\} \longrightarrow \mathsf{Y}_\phi^\bullet = \{[]\}$ obviously maps $[]$ to $[]$.

(2) If ω is an endotope, say $\omega = \mathsf{Y}_\psi$ for some $\psi \in \mathbb{O}_{n-1}$, then $\mathtt{t}\omega = \psi$. Further, $\omega^! = \{[[q]] \mid [q] \in \psi^\bullet\}$, and $\wp_\omega$ maps $[[q]]$ to $[q]$.

(3) Otherwise, ω decomposes as $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for some $\nu \in \mathbb{O}_n$, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$, and

$$\mathtt{t}\omega = (\mathtt{t}\nu) \underset{\wp_\nu[l]}{\square} \psi.$$

The readdressing function $\wp_\omega : \omega^! \longrightarrow (\mathtt{t}\omega)^\bullet$ is given as follows. Let $[j] \in \omega^!$.

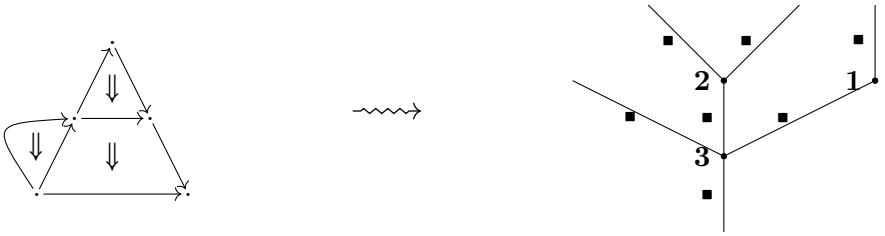
(a) If $[l] \sqsubseteq [j]$, then $[j] = [l[q]]$ for some $[q] \in \psi^\bullet$, and $\wp_\omega[l[q]] = (\wp_\nu[l]) \cdot [q]$.

(b) If $[l] \not\sqsubseteq [j]$, then $[j] \in \nu^!$. Assume $\wp_\nu[l] \sqsubseteq \wp_\nu[j]$. Then $\wp_\nu[j] = (\wp_\nu[l]) \cdot [[q]] \cdot [a]$, for some $[q] \in (\mathsf{s}_{\wp_\nu[l]}\mathtt{t}\nu)^\bullet = (\mathtt{t}\psi)^\bullet$, and let $\wp_\omega[j] = (\wp_\nu[l]) \cdot (\wp_\psi^{-1}[q]) \cdot [a]$.

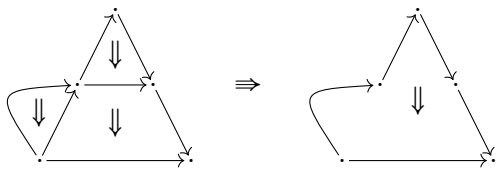
(c) If $\wp_\nu[l] \not\sqsubseteq \wp_\nu[j]$, then $\wp_\omega[j] = \wp_\nu[j]$.

Proof. Direct consequence of lemma A.7 and theorem A.14. □

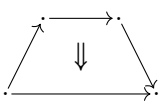
2.2.2 Opetopes vs. pasting diagrams Opetopes are closely related to the notion of *pasting diagram* commonly used in higher category theory to describe composition of higher cells. Informally, a pasting diagram of dimension n is a tree whose nodes are decorated with *n-cells*, edges with $(n-1)$-cells, and where the output edge of a node corresponds to its target, and the input edges to the cells in its source. For instance, the figure on the left below is a graphical representation of a 2-pasting diagram, and the corresponding decorated tree is drawn on the right:



If we consider the k-cells as k-opetopes for all k, then an n-pasting diagram P with $n \geq 1$ induces a $\mathfrak{Z}^{n-1}$-tree, i.e. a $(n+1)$-opetope, say ω. We say that P is the *source pasting diagram* of ω. The opetope ω also has a target $\mathtt{t}\omega \in \mathbb{O}_{n-1}$, and in the sequel, graphical representations of opetopes include both the source pasting diagram and the target. For instance, if P is the pasting diagram above, then ω is represented by



The dichotomy between pasting diagrams and opetopes can lead to ambiguities. For example,



represents a 2-opetope (the opetopic integer **3**), but also a 2-pasting diagram having a unique 2-cell. In this work, such ambiguities shall be resolved by the surrounding context.

2.2.3 Higher addresses By definition, an opetope ω of dimension $n \geq 2$ is a $\mathfrak{Z}^{n-2}$-tree, and thus the formalism of tree addresses (definition 2.8 and remark 2.31) can be applied to reference nodes of ω, also called its *source faces* or simply *sources*. In this section, we iterate this formalism into the concept of *higher dimensional address*.

Definition 2.38 (Higher address)**.** The set $\mathbb{A}_n$ of *n-addresses* is defined as follows. Let $\mathbb{A}_{-1} := \varnothing$, and for $n \geq 0$, let $\mathbb{A}_n$ be the set of bracket-enclosed sequences of elements of $\mathbb{A}_{n-1}$,

$$\mathbb{A}_{n+1} = \{[p] \mid p \in \mathbb{A}_n^*\}.$$

By convention, we also write the empty 0-address $[] \in \mathbb{A}_0$ as $*$. Note that the empty address $[]$, is in $\mathbb{A}_n$ for all $n \geq 0$. However, the surrounding context will always make the notation unambiguous.

Example 2.39. Since the unique element of $\mathbb{A}_0$ is $*$, the set of 1-addresses is

$$\mathbb{A}_1 = \{[\underbrace{* * \cdots *}_{k}] \mid k \in \mathbb{N}\}.$$

A 1-address $[* * \cdots *]$ where $*$ occurs k times will be more concisely written $[*^k]$. The following are higher addresses[4]:

$$[[][*][]] \in \mathbb{A}_2, \qquad [[[][*][**]][[* * *]]] \in \mathbb{A}_3, \qquad [[[[*]]]] \in \mathbb{A}_4.$$

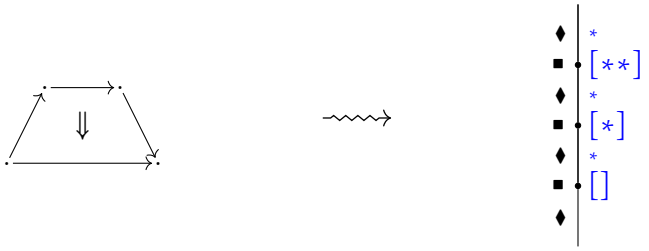
The expression $[*[*]]$ is not a valid higher address, as $*$ and $[*]$ do not have the same dimension.

For $\omega \in \mathbb{O}$ an opetope, nodes of ω can be specified uniquely using higher addresses, as we now show. Recall that E_{n-1} is the set of inputs of $\mathfrak{Z}^{n-2}$ (equation (2.34)). In $\mathfrak{Z}^0$, we set $E_1(\blacksquare) = \{*\}$, so that the unique "node address"[5] of $\blacksquare$ is $* \in \mathbb{A}_0$. Let $n \geq 2$, and assume by induction that for all $1 \leq k < n$ and all k-opetopes ψ, the nodes of ψ are assigned $(k-1)$-addresses, i.e. that we have an injective map $\& : \psi^\bullet \longrightarrow \mathbb{A}_{k-1}$. In particular, we consider the set $\psi^\bullet$ as a subset of $\mathbb{A}_{k-1}$. Recall that an opetope $\omega \in \mathbb{O}_n$ is a $\mathfrak{Z}^{n-2}$-tree $\omega : \langle\omega\rangle \longrightarrow \mathfrak{Z}^{n-2}$ (notation 2.7). Thus, addresses of nodes in ω are (bracket-enclosed) sequences of elements of E_{n-1}. By definition of $\mathfrak{Z}^{n-2}$ and by definition 2.29, we have

$$E_{n-1} = \bigcup_{\psi \in \mathbb{O}_{n-1}} \psi^\bullet \subseteq \mathbb{A}_{n-2}.$$

In conclusion, node and edge addresses of ω are $(n-1)$-addresses.

Example 2.40. Consider the 2-opetope on the left, called **3**:

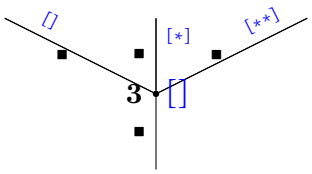


[4] Ambiguity with the dimension of addresses could be lifted altogether by indicating the dimension as e.g. $[]^1 \in \mathbb{A}_1$, $[* * * *]^1 \in \mathbb{A}_1$, $[[]^1[*]^1[]^1]^2 \in \mathbb{A}_2$, $[[[[*]^1]^2]^3]^4 \in \mathbb{A}_4$. However, this makes notations significantly heavier, so we avoid using this convention.

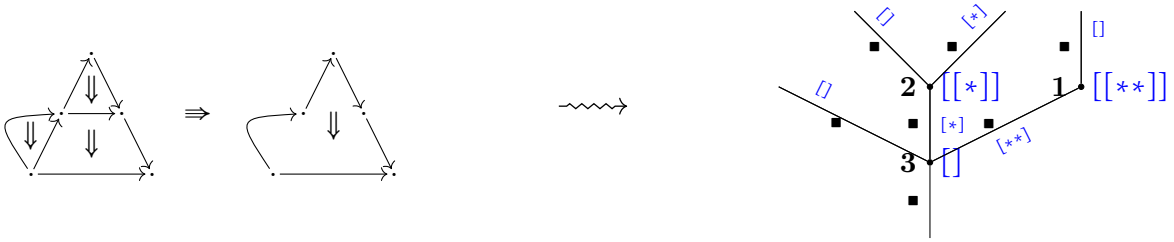
[5] Of course, $\blacksquare$ is not a tree, but this abuse of terminology is convenient, as it allows us to talk about higher addresses and opetopes in a more uniform manner.

Its underlying pasting diagram consists of 3 arrows ▪ grafted linearly. Since the only node address of ▪ is $* \in \mathbb{A}_0$, the underlying tree of **3** can be depicted as on the right. On the left of that tree are the decorations: nodes are decorated with ▪ $\in \mathbb{O}_1$, while edges are decorated with ♦ $\in \mathbb{O}_0$. For each node in the tree, the set of input edges of that node is in bijective correspondence with the node addresses of the decorating opetope, written on the right of each edge. In this low dimensional example, those addresses can only be $*$. Finally, on the right of each node of the tree is its 1-address, which is just a sequence of 0-addresses giving "walking instructions" to get from the root to that node.

The 2-opetope **3** can then be seen as a corolla in some 3-opetope as follows:



As previously mentioned, the set of input edges is in bijective correspondence with the set of node addresses of **3**. Here is now an example of a 3-opetope (already studied in section 2.2.2), with its annotated underlying tree on the right (the 2-opetopes **1** and **2** are analogous to **3**):



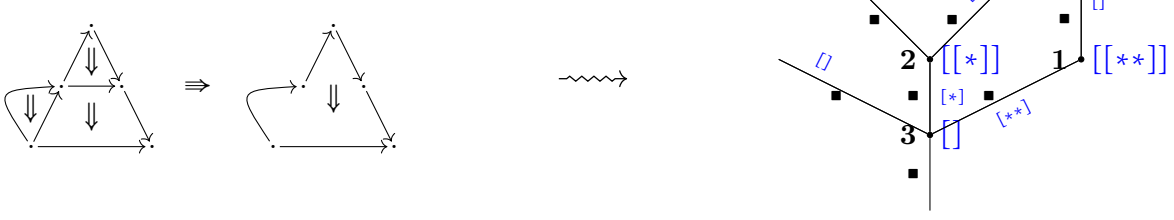
Further, the leaf addresses of this opetope are [[]], [[*][]], [[*][*]], and [[**][]].

Definition 2.41. (1) If $[p_1], [p_2] \in \mathbb{A}_n$, then their *concatenation* is $[p_1] \cdot [p_2] := [p_1 p_2]$. In particular, $[p_1] \cdot [] = [] \cdot [p_1] = [p_1]$.

(2) Let $\sqsubseteq$ be the *prefix order* on $\mathbb{A}_n$, i.e. $[p_1] \sqsubseteq [p_2]$ if and only if there exists $[p_3] \in \mathbb{A}_n$ such that $[p_1] \cdot [p_3] = [p_2]$.

(3) Let $\preceq$ be the *lexicographical order* on $\mathbb{A}_n$. It is trivial on $\mathbb{A}_0$, given by the prefix order on $\mathbb{A}_1$, and on $\mathbb{A}_n$, it is induced by lexicographical order on $\mathbb{A}_{n-1}$.

Example 2.42. Consider the 3-opetope of example 2.40:



On nodes, we have the ordering $[] \prec [[*]] \prec [[**]]$, and on edges, $[] \prec [[]] \prec [[*]] \prec [[*][]] \prec [[*][*]] \prec [[**]] \prec [[**][]]$.

2.2.4 The category of opetopes In this subsection, we define the category $\mathbb{O}$ of opetopes. We first begin by a result whose geometrical meaning is explained later in this section.

Lemma 2.43 (Opetopic identities)**.** *Let $\omega \in \mathbb{O}_n$ with $n \geq 2$.*

Inner edge. For an inner edge $[p[q]] \in \omega^\bullet$ (the fact that ω has an inner edge implies that it is non-degenerate), we have $\mathsf{t}\mathsf{s}_{[p[q]]}\,\omega = \mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega$.

Globularity 1. If ω is non-degenerate, we have $\mathsf{t}\mathsf{s}_{[]}\,\omega = \mathsf{t}\mathsf{t}\omega$.

Globularity 2. If ω is non-degenerate, and $[p[q]] \in \omega^!$, we have $\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega = \mathsf{s}_{\wp\omega[p[q]]}\,\mathsf{t}\omega$.

Degeneracy. If ω is degenerate, we have $\mathsf{s}_{[]}\,\mathsf{t}\omega = \mathsf{t}\mathsf{t}\omega$.

Proof. *Inner edge.* By definition of a $\mathfrak{Z}^{n-2}$-tree.

Globularity 1 and 2. By theorem 2.26, the monad structure on $\mathfrak{Z}^{n-2}$ amounts to a structure map $(\mathfrak{Z}^{n-2})^\star \longrightarrow \mathfrak{Z}^{n-2}$, which, taking the notations of definition 2.27, is written as

$$\begin{array}{ccccccc}
\mathbb{O}_{n-2} & \xleftarrow{\mathsf{e}} & \mathrm{tr}^!\,\mathfrak{Z}^{n-2} & \xrightarrow{p} & \mathrm{tr}\,\mathfrak{Z}^{n-2} & \xrightarrow{\mathsf{e}_{[]}} & \mathbb{O}_{n-2} \\
\| & & \downarrow{\scriptstyle \wp} & \lrcorner & \downarrow{\scriptstyle \mathsf{t}} & & \| \\
\mathbb{O}_{n-2} & \xleftarrow{\mathsf{s}} & \mathbb{O}^\bullet_{n-1} & \xrightarrow{p} & \mathbb{O}_{n-1} & \xrightarrow{\mathsf{t}} & \mathbb{O}_{n-2}.
\end{array}$$

The claims follow from the commutativity of the right and left square respectively.

Degeneracy. Let $\omega = \mathsf{l}_\phi$, for $\phi \in \mathbb{O}_{n-2}$. By proposition 2.37, $\mathsf{t}\mathsf{t}\omega = \mathsf{t}\mathsf{Y}_\phi = \phi$, and clearly, $\phi = \mathsf{s}_{[]}\,\mathsf{Y}_\phi = \mathsf{s}_{[]}\,\mathsf{t}\omega$. □

Definition 2.44 (The category $\mathbb{O}$ of opetopes)**.** The identities of lemma 2.43 allow us to define the category $\mathbb{O}$ of opetopes by generators and relations as follows.

Objects. We set $\mathrm{ob}\,\mathbb{O} = \sum_{n\in\mathbb{N}} \mathbb{O}_n$.

Generating morphisms. Let $\omega \in \mathbb{O}_n$ with $n \geq 1$. We introduce a generator $\mathsf{t} : \mathsf{t}\omega \longrightarrow \omega$, called the *target embedding*. If $[p] \in \omega^\bullet$, then we introduce a generator $\mathsf{s}_{[p]} : \mathsf{s}_{[p]}\,\omega \longrightarrow \omega$, called a *source embedding*. An *elementary face embedding* is either a source or the target embedding.

Relations. We impose 4 relations described by the following commutative squares, which just enforce the identities of lemma 2.43. Let $\omega \in \mathbb{O}_n$ with $n \geq 2$.

(Inner) For $[p[q]] \in \omega^\bullet$ (forcing ω to be non-degenerate), the following square must commute:

$$\begin{array}{ccc}
\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega & \xrightarrow{\mathsf{s}_{[q]}} & \mathsf{s}_{[p]}\,\omega \\
{\scriptstyle \mathsf{t}}\downarrow & & \downarrow{\scriptstyle \mathsf{s}_{[p]}} \\
\mathsf{s}_{[p[q]]}\,\omega & \xrightarrow{\mathsf{s}_{[p[q]]}} & \omega
\end{array}$$

(Glob1) If ω is non-degenerate, the following square must commute:

$$\begin{array}{ccc}
\mathsf{t}\mathsf{t}\omega & \xrightarrow{\mathsf{t}} & \mathsf{t}\omega \\
{\scriptstyle \mathsf{t}}\downarrow & & \downarrow{\scriptstyle \mathsf{t}} \\
\mathsf{s}_{[]}\,\omega & \xrightarrow{\mathsf{s}_{[]}} & \omega.
\end{array}$$

(Glob2) If ω is non-degenerate, and for $[p[q]] \in \omega^!$, the following square must com-

mute:

$$\begin{array}{ccc} \mathsf{s}_{\wp_\omega[p[q]]}\,\mathsf{t}\,\omega & \xrightarrow{\mathsf{s}_{\wp_\omega[p[q]]}} & \mathsf{t}\,\omega \\ {\scriptstyle \mathsf{s}_{[q]}}\downarrow & & \downarrow{\scriptstyle \mathsf{t}} \\ \mathsf{s}_{[p]}\,\omega & \xrightarrow{\mathsf{s}_{[p]}} & \omega. \end{array}$$

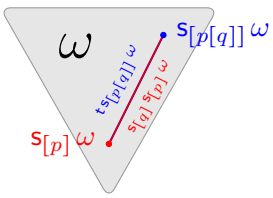
(Degen) If ω is degenerate, the following square must commute:

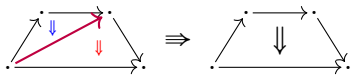
$$\begin{array}{ccc} \mathsf{t}\,\mathsf{t}\,\omega & \xrightarrow{\mathsf{t}} & \mathsf{t}\,\omega \\ {\scriptstyle \mathsf{s}_{[]}}\downarrow & & \downarrow{\scriptstyle \mathsf{t}} \\ \mathsf{t}\,\omega & \xrightarrow{\mathsf{t}} & \omega. \end{array}$$

Definition 2.45. An *opetopic set* is a $\mathcal{S}$et-valued presheaf over $\mathbb{O}$. We write $\widehat{\mathbb{O}}$ for the category of opetopic sets and natural transformations, $O[-] : \mathbb{O} \longrightarrow \widehat{\mathbb{O}}$ for the Yoneda embedding, and $\widehat{\mathbb{O}}_{\mathrm{fin}}$ for the full subcategory of $\widehat{\mathbb{O}}$ spanned by finite opetopic sets, i.e. those $X \in \widehat{\mathbb{O}}$ such that $\sum_{\omega \in \mathbb{O}} X_\omega$ is a finite set. Equivalently, $\widehat{\mathbb{O}}_{\mathrm{fin}}$ is the completion of $\mathbb{O}$ under finite colimits.

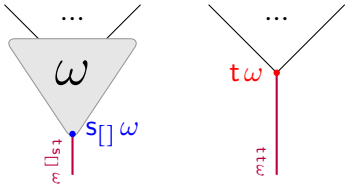
Remark 2.46. Let us explain this definition a little more. Opetopes are trees whose nodes (and edges) are decorated by opetopes. The decoration is now interpreted as a geometrical feature, namely as an embedding of a lower dimensional opetope. Further, the target of an opetope, while not an intrinsic data, is also represented as an embedding. The relations can be understood as follows.

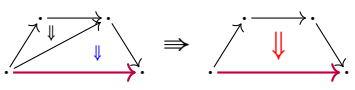
(Inner) The inner edge at $[p[q]] \in \omega^\bullet$ is decorated by the target of the decoration of the node "above" it (here $\mathsf{s}_{[p[q]]}\,\omega$), and in the $[q]$-source of the node "below" it (here $\mathsf{s}_{[p]}\,\omega$). By construction, those two decorations match, and this relation makes the two corresponding embeddings $\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega \longrightarrow \omega$ match as well. On the left is an informal diagram about ω as a tree (reversed gray triangle), and on the right is an example of pasting diagram represented by an opetope, with the relevant features of the **(Inner)** relation colored or thickened.

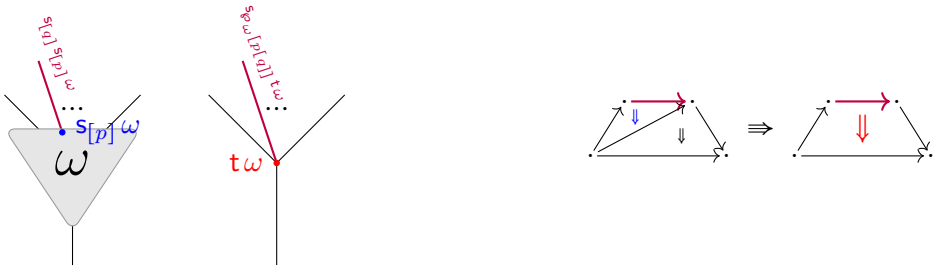




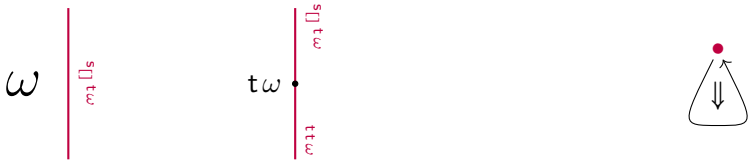
(Glob1-2) If we consider the underlying tree of ω as its "geometrical source", and the corolla $\mathsf{Y}_{\mathsf{t}\omega}$ as its "geometrical target", then they should be parallel. The relation **(Glob1)** expresses this idea by "gluing" the root edges of ω and $\mathsf{Y}_{\mathsf{t}\omega}$ together, while **(Glob2)** glues the leaves according to $\wp_\omega$.







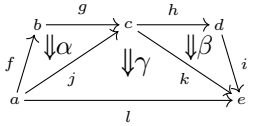
(Degen) If ω is a degenerate opetope, depicted as on the right, then its target should be a "loop", i.e. its only source and its target should be glued together.



3. Named approach

3.1 The system for opetopes

3.1.1 Syntax In this section, we define the underlying syntax of Opt$^!$, our named derivation system for opetopes. As explained in the introduction, a typical pasting diagram is pictured below:



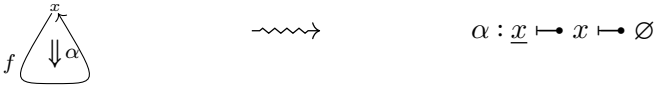
We shall use the names of the cells of this picture as variables, and encode the pasting diagram as the following expression:

$$\gamma(j \leftarrow \alpha, k \leftarrow \beta).$$

Here, j, k, α, β, and γ are now variables, equipped with a dimension (1 for j and k, and 2 for α , β, and γ), and the notation is meant to be read as "the variable γ in which α (resp. β) has been formally grafted on the input labeled by j (resp. k)". Such a term will be given a type:

$$i(t \leftarrow h(z \leftarrow g(y \leftarrow f))) \mapsto a \mapsto \varnothing,$$

which expresses the fact that the source is the "composite" $i \cdot h \cdot g \cdot f$, and that the source of the source is a. Since the pasting diagram is 2-dimensional, there is no further iterated source, and we conclude the sequence by a $\varnothing$ symbol. Similarly, the degenerate pasting diagram on the left below will be denoted by the typed term figured on the right:



where the term $\underline{x}$ denotes a degenerate 1-dimensional pasting diagram with x as source.

Definition 3.1 (Term)**.** We assume that we have an $\mathbb{N}$-graded set $\mathbb{V}$ of variables. Elements of $\mathbb{V}_n$ represent n-dimensional cells. An *n-term* is constructed according to the following grammar:

$$\begin{array}{rcll} \mathbb{T}_{-1} & ::= & \{\varnothing\} & \text{by convention} \\ \mathbb{T}_0 & ::= & \mathbb{V}_0 & \\ \mathbb{T}_{n+1} & ::= & \underline{\mathbb{V}_n} \quad | \quad \mathbb{T}'_{n+1} & \\ \mathbb{T}'_{n+1} & ::= & \mathbb{V}_{n+1}(\mathbb{V}_n \leftarrow \mathbb{T}'_{n+1}, \ldots) & \end{array}$$

where the expression "$\mathbb{V}_n \leftarrow \mathbb{T}'_{n+1}, \ldots$" signifies that there is 0 or more instances of the "$\mathbb{V}_n \leftarrow \mathbb{T}'_{n+1}$" part between the parentheses. Terms of the form $\underline{u}$ are called *degenerate terms*, or *empty* syntactic pasting diagrams. Note that there are no degenerate 0-terms.

Example 3.2. If $f, g \in \mathbb{V}_1$, and $a \in \mathbb{V}_0$, then the following is an element of $\mathbb{T}_1$:

$$g(a \leftarrow f())$$

To make notations lighter, we omit empty parentheses "()", so the previous 1-term can be more concisely written as $g(a \leftarrow f)$. Since $f \in \mathbb{V}_1$, the expression $\underline{f}$ is a degenerate term in $\mathbb{T}_2$.

Notation 3.3. A term of the form $g(a_1 \leftarrow f_1, \ldots, a_k \leftarrow f_k)$ will oftentimes be abbreviated as $g(\overrightarrow{a_i \leftarrow f_i})$, leaving k implicit. By convention, the sequence $a_1 \leftarrow f_1, \ldots, a_k \leftarrow f_k$ above is always considered up to permutation, i.e. for σ a bijection of the set $\{1, \ldots, k\}$, the terms $g(a_1 \leftarrow f_1, \ldots, a_k \leftarrow f_k)$ and $g(a_{\sigma(1)} \leftarrow f_{\sigma(1)}, \ldots, a_{\sigma(k)} \leftarrow f_{\sigma(k)})$ are considered equal.

Notation 3.4. For $t \in \mathbb{T}_n$, write $t^\bullet$ for the set of n-variables occurring in t. In the previous example, $(g(a \leftarrow f))^\bullet = \{f, g\}$. Note that $x \in x^\bullet$ for all $x \in \mathbb{V}_n$.

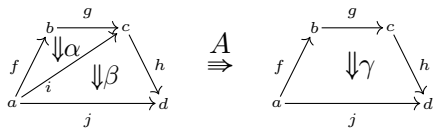
Definition 3.5 (Type)**.** An *n-type* T is a sequence of terms of the form

$$s_1 \mapsto\bullet\ s_2 \mapsto\bullet\ \cdots \mapsto\bullet\ s_n \mapsto\bullet\ \varnothing, \tag{3.6}$$

where $s_i \in \mathbb{T}_{n-i}$. As we will see in proposition 3.53 (see also remark 4.18), the $\mapsto\bullet$ symbols hints the existence of a bijection between the "leaves" of the term on its left and the "nodes" of the term on its right.

Definition 3.7 (Typing)**.** A *typing* of a term $t \in \mathbb{T}_n$ is an expression of the form $t : T$, for T an n-type. If T is as in equation (3.6), then s_i is thought of as the i-th (iterated) source of t. We then write $\mathsf{s}\, t := s_1$ (which tacitly depends on the type T), and more generally $\mathsf{s}^i\, t := s_i$. By convention, $\mathsf{s}^0\, t := t$.

Example 3.8. The pasting diagram on top will be described by the typing below:



$$\rightsquigarrow \qquad A : \underbrace{\beta(i \leftarrow \alpha)}_{=\mathsf{s}\,A} \mapsto\bullet\ \underbrace{h(c \leftarrow g(b \leftarrow f))}_{=\mathsf{s}^2\,A} \mapsto\bullet\ \underbrace{a}_{=\mathsf{s}^3\,A} \mapsto\bullet\ \underbrace{\varnothing}_{=\mathsf{s}^4\,A}.$$

Remark that the term $\mathsf{s}^2\, A = h(c \leftarrow g(b \leftarrow f))$ is the source pasting diagram of the *target* of A, as in $\mathsf{ss}\, A = \mathsf{st}\, A$. Targets are not represented in this syntax, but already, we see that the information they carry is not lost (here, γ does not appear in the syntactical description of A, but it source does). Refer to proposition 3.53 for a formal account of this observation.

Definition 3.9 (Context)**.** A *context* Γ is a set of typings of distinct variables, and is more commonly written as a list.

Notation 3.10. Write $\mathbb{V}_{\Gamma,k}$ for the set of k-variables typed in Γ, let $\mathbb{V}_\Gamma := \sum_{k\in\mathbb{N}} \mathbb{V}_{\Gamma,k}$, write $\mathbb{T}_{\Gamma,k}$ for the set of k-terms whose variables (in any dimension) are in $\mathbb{V}_\Gamma$, and $\mathbb{T}_\Gamma := \sum_{k\in\mathbb{N}} \mathbb{T}_{\Gamma,k}$.

Remark 3.11. As we will see (inference rules in definition 3.23), for a derivable context Γ, if x occurs in the typing of a variable of Γ, then $x \in \mathbb{V}_\Gamma$. Note that in any context Γ, if a variable $x \in \mathbb{V}_{\Gamma,k}$ occurs in the type of $y \in \mathbb{V}_{\Gamma,l}$, then $k < l$. In particular, there is no cyclic dependency among variables.

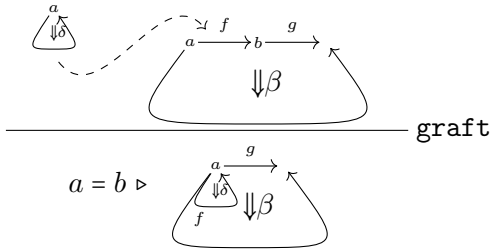
Definition 3.12 (Equational theory)**.** Let Γ be a context. An *equational theory* E on $\mathbb{V}_\Gamma$ is a set of formal equalities between variables of Γ. We write $=_E$ for the equivalence relation on $\mathbb{V}_\Gamma$ generated by E (or just $=$ if the E is clearly implied).

Definition 3.13 (Sequent)**.** A *sequent* is an expression of the form

$$E \triangleright \Gamma \vdash t : T$$

where Γ is a context, E is an equational theory on $\mathbb{V}_\Gamma$, and the right hand side is a typing. We may write $\vdash_n$ to signify that $t \in \mathbb{T}_n$. The equivalence relation $=_E$ on $\mathbb{V}_\Gamma$ extends to $\mathbb{T}_\Gamma$ in an obvious way. If the equational theory E is clear, and if $x =_E y$ with $y \in t^\bullet$, then by convention, we consider that $x \in t^\bullet$, so that x and y really are interchangeable.

Remark 3.14. As illustrated below, grafting degenerate terms produces identifications of lower dimensional variables, which must be accounted for. This will be the role of the equational theories.



This example will be treated in detail in example 3.69.

Definition 3.15 (Equivalence of sequents)**.** If $(E \triangleright \Gamma \vdash t : T)$ and $(F \triangleright \Upsilon \vdash u : U)$ are sequents, and if there exists a bijection $\sigma : \mathbb{V}_\Upsilon \longrightarrow \mathbb{V}_\Gamma$ such that

$$(E \triangleright \Gamma \vdash t : T) \;=\; (F^\sigma \triangleright \Upsilon^\sigma \vdash u^\sigma : U^\sigma)\,,$$

where $(-)^\sigma$ is the substitution according to σ, then we say that both sequents are *α-equivalent* (or just *equivalent*), denoted by

$$(E \triangleright \Gamma \vdash t : T) \simeq (F \triangleright \Upsilon \vdash u : U)\,.$$

Definition 3.16 (Leaves of a term)**.** Let Γ be a context and $t \in \mathbb{T}_{\Gamma,n}$ be an n-term such that all variables in $t^\bullet$ are typed in Γ. The set $t^!$ of *leaves* of t is defined as follows.

(1) If $n = 0$, then $t^! = \varnothing$.

(2) If $t = \underline{a}$ is degenerate, with $a \in \mathbb{V}_{\Gamma,n-1}$, then $t^! = \{a\}$.

(3) If $t \in \mathbb{V}_{\Gamma,n}$ is a variable, then by assumption, it is typed in Γ, and thus has a source $\mathtt{s}t$. We define $t^! := (\mathtt{s}t)^\bullet$.
(4) Otherwise, t decomposes as $t = x(\overrightarrow{y_i \leftarrow u_i})$, where $x \in \mathbb{V}_{\Gamma,n}$, $y_i \in \mathbb{V}_{\Gamma,n-1}$, and $u_i \in \mathbb{T}_{\Gamma,n}$. We define

$$t^! := (x^! - \{y_1, \ldots\}) \bigcup_i u_i^!.$$

Note that without any assumption, the set $\{y_1, \ldots\}$ is not necessarily a subset of $x^!$, and so by $x^! - \{y_1, \ldots\}$ we of course mean $x^! - (x^! \cap \{y_1, \ldots\})$.

Definition 3.17 (Well-formed term)**.** Let Γ be a context. We define the set of *well-formed terms* inductively as follows.

(1) Degenerate terms are well-formed.
(2) Variables are well-formed terms.
(3) Consider a term $t \in \mathbb{T}_{\Gamma,n}$ of the form $x(\overrightarrow{y_i \leftarrow u_i})$, where $x \in \mathbb{V}_{\Gamma,n}$, $y_i \in \mathbb{V}_{\Gamma,n-1}$, and $u_i \in \mathbb{T}_{\Gamma,n}$. Then t is well-formed if
   (a) x is typed in Γ, and $\{y_1, \ldots\} \subset x^!$;
   (b) the u_is are themselves well-formed.

   Note that if t is well-formed, then all variables in $t^\bullet$ are typed in Γ.

3.1.2 Inference rules We present the inference rules of the OPT$^!$ system in definition 3.23. Rule `graft` requires the so-called graft notation and substitution operation, which we introduce now.

Definition 3.18 (Grafting of terms)**.** Let Γ be a context and E be an equational theory on $\mathbb{V}_\Gamma$. For $t \in \mathbb{T}_{\Gamma,n}$ a well-formed term, $a \in \mathbb{V}_{n-1}$, and $x \in \mathbb{V}_{\Gamma,n}$, the graft notation $t(a \leftarrow x)$ can be iteratively simplified depending on the structure of t, according to the following rewriting rules.

(1) If t is degenerate, say $t = \underline{b}$, then

$$t(a \leftarrow x) \quad \longmapsto \quad \begin{cases} t & \text{if } a \neq_E b, \\ x & \text{if } a =_E b. \end{cases}$$

(2) If t is not degenerate, then we can write it as $t = y(\overrightarrow{z_i \leftarrow u_i})$ with $y \in \mathbb{V}_{\Gamma,n}$, $z_i \in \mathbb{V}_{\Gamma,n-1}$ and $u_i \in \mathbb{T}_{\Gamma,n}$. Then,

$$t(a \leftarrow x) \quad \longmapsto \quad \begin{cases} y(\overrightarrow{z_i \leftarrow u_i(a \leftarrow x)}) & \text{if } a \notin y^!, \\ y(\overrightarrow{z_i \leftarrow u_i}, a \leftarrow x) & \text{if } a \in y^!. \end{cases} \tag{3.19}$$

In particular, note that if $a \notin y^!$, then $y()(a \leftarrow x)$ rewrites to $y()$, and with the "empty parentheses convention", this gives $y(a \leftarrow x) \longmapsto y$.

Clearly, in both cases, the resulting term is well-formed.

Lemma 3.20. *Let Γ be a context and E be an equational theory on $\mathbb{V}_\Gamma$, $t \in \mathbb{T}_{\Gamma,n}$ be a well-formed term, $a \in \mathbb{V}_{n-1}$, and $x \in \mathbb{V}_{\Gamma,n}$ be a variable typed in Γ. If $a \in t^!$, then*

$$t(a \leftarrow x)^! = (t^! - \{a\}) \cup x^!.$$

Proof. We proceed by induction.

(1) Assume that t is degenerate, say $t = \underline{b}$ for some $b \in \mathbb{V}_{\Gamma,n-1}$. Since $a \in t^! = \{b\}$, we have $a = b$, and then by definition, $t(a \leftarrow x)$ rewrites to x. The result follows trivially.

(2) Assume that $t \in \mathbb{V}_{\Gamma,n}$ is a variable. Since $a \in t^!$, $t(a \leftarrow x)$ can only rewrite to $t(a \leftarrow x)$, i.e. the $a \leftarrow x$ expression is not discarded. The result follows by definition 3.16.

(3) Assume that $t \in \mathbb{V}_{\Gamma,n}$ is not degenerate nor a variable, and decompose it as $t = y(\overrightarrow{z_i \leftarrow u_i})$ with $y \in \mathbb{V}_{\Gamma,n}$, $z_i \in \mathbb{V}_{\Gamma,n-1}$ and $u_i \in \mathbb{T}_{\Gamma,n}$. By definition 3.18, $t(a \leftarrow x)$ rewrites to

$$y(\overrightarrow{z'_j \leftarrow u'_j}) = \begin{cases} y(\overrightarrow{z_i \leftarrow u_i(a \leftarrow x)}) & \text{if } a \notin y^!, \\ y(\overrightarrow{z_i \leftarrow u_i}, a \leftarrow x) & \text{if } a \in y^!. \end{cases}$$

However, since $a \in t^! = (\mathtt{s}t)^\bullet$, $x \in t(a \leftarrow x)^\bullet$, or in other words, the expression $a \leftarrow x$ is not discarded. In particular, there exists at least one index k such that $x \in u'_k(a \leftarrow x)^\bullet$. Let K be the set of such indices. We now distinguish cases:

(a) If $a \notin y^!$, then $z'_i = z_i$ for all i, $u'_i = u_i$ for $i \notin K$, and $u'_k = u_k(a \leftarrow x)$ for all $k \in K$. Then,

$$\begin{aligned} t(a \leftarrow x)^! &= y(\overrightarrow{z'_j \leftarrow u'_j})^! \\ &= \left((y^! - \{z_1, \ldots\}) \bigcup_{i \notin K} u_i^! \right) \bigcup_{k \in K} u_k(a \leftarrow x)^! \\ &= \left((y^! - \{z_1, \ldots\}) \bigcup_{i \notin K} u_i^! \right) \bigcup_{k \in K} (u_k^! - \{a\}) \cup x^! \qquad \text{by induction} \\ &= \left(\left((y^! - \{z_1, \ldots\}) \bigcup_i u_i^! \right) - \{a\} \right) \cup x^! \\ &= (t^! - \{a\}) \cup x^!. \end{aligned}$$

(b) If $a \in y^!$, then $t(a \leftarrow x) = y(\overrightarrow{z_i \leftarrow u_i}, a \leftarrow x)$, and

$$\begin{aligned} t(a \leftarrow x)^! &= (x^! - \{a, y_1, \ldots\}) \cup x^! + \bigcup_i u_i^! \\ &= (t^! - \{a\}) \cup x^!. \end{aligned}$$

□

Definition 3.21 (Substitution in terms)**.** Let Γ be a context, E be an equational theory on $\mathbb{V}_\Gamma$, $u, v \in \mathbb{T}_{\Gamma,n}$ be well-formed terms, and $a \in \mathbb{V}_{\Gamma,n-1}$ be a typed variable such that $a \in u^\bullet$. Assume that $v^! = a^!$. The *substitution* of a for v in the term u, denoted by $u[v/a]$, is defined as follows. If u is degenerate, then $u[v/a] := u$. Otherwise, u decomposes as $u = y(\overrightarrow{z_i \leftarrow w_i})$ with $y \in \mathbb{V}_{\Gamma,n}$, $z_i \in \mathbb{V}_{\Gamma,n-1}$ and $w_i \in \mathbb{T}_{\Gamma,n}$.

(Subst1) If v is not degenerate, then

$$u[v/a] := \begin{cases} y(\overrightarrow{z_i \leftarrow w_i[v/a]}) & \text{if } a \neq_E y, \\ v(\overrightarrow{z_i \leftarrow w_i}) & \text{if } a =_E y, \end{cases} \tag{3.22}$$

and in the second case, definition 3.18 is applied to evaluate the resulting term.

(Subst2) If v is degenerate, say $v = \underline{b}$ for $b \in \mathbb{V}_{\Gamma,n-1}$, then, $u[\underline{b}/a]$ is defined by cases on the form of u:

(Subst2a) if $u =_E a$, then $u[\underline{b}/a] := \underline{b}$;

(Subst2b) if u is of the form $a(b_1 \leftarrow r_1, \ldots, b_k \leftarrow r_k)$, then since $\{b_1, \ldots, b_k\} \subseteq a^! = v^! = \underline{b}^! = \{b\}$, we have $k = 1$ and $b_1 = b$, and we define $u[\underline{b}/a] := r_1$;

(Subst2c) if u is of the form $y(\dots, z \leftarrow a, \dots)$, then we remove the grafting $z \leftarrow a$, and we add the equality $b = z$ to the ambient equational theory as a *side effect*;

(Subst2d) if u is of the form $y(\dots, z \leftarrow a(b_1 \leftarrow r_1, \dots, b_k \leftarrow r_k), \dots)$, then following the same argument as in **(Subst2b)**, $k = 1$, $b_1 = b$, and

$$u[\underline{b}/a] := y(\dots, z \leftarrow r_1, \dots),$$

but we also add the equality $b = z$ to the ambient equational theory as a *side effect*;

(Subst2e) otherwise, if u is of the form $y(\overrightarrow{z_i \leftarrow w_i})$, and if the previous cases do not apply (i.e. a is not the front variable of u or w_i for all i), then we recursively evaluate the substitution:

$$u[\underline{b}/a] := y(\overrightarrow{z_i \leftarrow w_i[\underline{b}/a]}).$$

We emphasize that cases **(Subst2c)** and **(Subst2d)** potentially incur a side effect on the ambient equational theory. In each case, it is straightforward to check that the resulting term is well-formed. Further, note that $u[v/a]^\bullet = (u^\bullet - \{a\}) \cup v^\bullet$.

Definition 3.23 (The Opt$^!$ system)**.** *Introduction of points.* This rule introduces 0-cells, also called points. If $x \in \mathbb{V}_0$, then

$$\frac{}{\triangleright x : \varnothing \vdash_0 x : \varnothing}\ \texttt{point}$$

Introduction of degeneracies. This rule derives empty pasting diagrams. If $x \in \mathbb{V}_n$, then

$$\frac{E \triangleright \Gamma \vdash_n x : T}{E \triangleright \Gamma \vdash_{n+1} \underline{x} : x \mapsto T}\ \texttt{degen}$$

Shift to the next dimension. This rule takes a term t and introduces a new cell x having t as source. If $x \in \mathbb{V}_{n+1} - \mathbb{V}_\Gamma$, then

$$\frac{E \triangleright \Gamma \vdash_n t : T}{E \triangleright \Gamma, x : t \mapsto T \vdash_{n+1} x : t \mapsto T}\ \texttt{shift}$$

Grafting. This rule glues an n-cell x onto an n-term t along a variable $a \in s_1^\bullet := (\mathtt{s}\, t)^\bullet$. We assume that Γ and Υ are compatible, in that for all $y \in \mathbb{V}$, if $y \in \mathbb{V}_\Gamma \cap \mathbb{V}_\Upsilon$, then the typing of y in both contexts match modulo the equational theory $E \cup F$. Further, the only variables typed in both Γ and Υ are the variables occurring in the sources of a (i.e. $\mathtt{s}^i\, a$, for $1 \le i \le n-1$). If $x \in \mathbb{V}_n$, $t \in \mathbb{T}_n$ is not degenerate, $a \in (\mathtt{s}\, t)^\bullet$ [6] is such that $\mathtt{s}\, a =_{E \cup F} \mathtt{ss}\, x$ [7], then

$$\frac{E \triangleright \Gamma \vdash_n t : s_1 \mapsto s_2 \mapsto \cdots \qquad F \triangleright \Upsilon \vdash_n x : U}{G \triangleright \Gamma \cup \Upsilon \vdash_n t(a \leftarrow x) : s_1[\mathtt{s}x/a] \mapsto s_2 \mapsto \cdots}\ \texttt{graft}$$

where the grafting $t(a \leftarrow x)$ is simplified as in definition 3.18, $s_1[\mathtt{s}x/a]$ is defined in definition 3.21, and where G is the union of E and F, and potentially a set of additional equalities incurred by the evaluation of $s_1[\mathtt{s}x/a]$. We also write $\texttt{graft-}a$ to make explicit that we grafted onto a.

Remarks 3.24. (1) Note that for each rule, the conclusion is still a sequent, in that variables in the context are only typed once. Further, it is easy to see that derivable sequents necessarily type well-formed terms. In particular, all terms occurring on either side of the turnstile $\vdash$ must be well-formed.

(2) In the `graft` rule, by minimality of the intersection of Γ and Υ, we can conclude that $t^! = s_1^\bullet$ and $x^! = (\mathtt{s}\,x)^\bullet$ are disjoint. We can then improve lemma 3.20:

$$t(a \leftarrow x)^! = s_1[\mathtt{s}\,x/a]^\bullet = s_1^\bullet - \{a\} + (\mathtt{s}\,x)^\bullet = t^! - \{a\} + x^! \tag{3.25}$$

(3) From the formulation of system Opt$^!$, it is clear that a sequent that is equivalent to a derivable one is itself derivable. Let us now turn our attention to rule `shift` above. It takes a term t, thought of as a pasting diagram, and creates a new variable having t as source. One may thus think of it as a rule creating "fillers", akin to Kan filler condition on simplicial sets.

Lemma 3.26. *Let $n \geq 1$. For $(E \triangleright \Gamma \vdash_n t : s_1 \longmapsto \cdots)$ a derivable sequent, we have $t^! = s_1^\bullet$.*

Proof. (1) It t is degenerate, say $t = \underline{x}$ for $x \in \mathbb{V}_{\Gamma,n-1}$, then the last rule in the proof tree of $(E \triangleright \Gamma \vdash_n t : s_1 \longmapsto \cdots)$ was an instance of the `degen` rule, and

$$\begin{aligned} t^! &= \underline{x}^! \\ &= \{x\} && \text{by definition 3.16} \\ &= x^\bullet. \end{aligned}$$

(2) If t is variable, then since $n \geq 1$, the last step in the proof tree was an instance of the `shift` rule, and the result is tautological.

(3) Otherwise, the last step in the proof tree was an instance of the `graft` rule. For convenience, let us change notations: consider the following instance

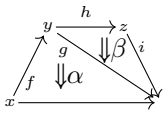
$$\frac{E \triangleright \Gamma \vdash_n t : s_1 \longmapsto \cdots \qquad F \triangleright \Upsilon \vdash_n x : u_1 \longmapsto \cdots}{G \triangleright \Gamma \cup \Upsilon \vdash_n t(a \leftarrow x) : s_1[u_1/a] \longmapsto \cdots}\ \mathtt{graft}$$

and assume that the result holds for the premise sequents, i.e. that $t^! = s_1^\bullet$ and $x^! = u_1^\bullet$ (which again, is a tautology, since x is a variable). We show that $t(a \leftarrow x)^! = s_1[u_1/a]^\bullet$:

$$\begin{aligned} t(a \leftarrow x)^! &= t^! - \{a\} + x^! && \text{by (3.25)} \\ &= s_1^\bullet - \{a\} + u_1^\bullet && \text{by inductive assumption} \\ &= s_1[u_1/a]^\bullet && \text{by definition 3.21.} \end{aligned}$$

□

Example 3.27. Consider the term $t = \alpha(g \leftarrow \beta)$ in a suitable context Γ:



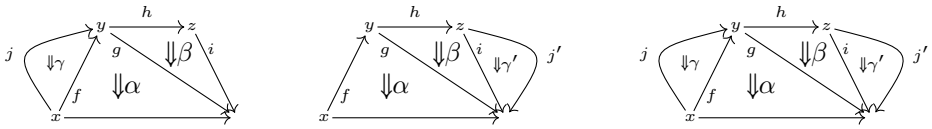
[6]This ensures that a has not been used for grafting beforehand.

[7]Recall that by example 3.8, we can understand this condition as $\mathtt{s}\,a = \mathtt{s}\,\mathtt{t}\,x$, so that the variable x may indeed be glued onto a.

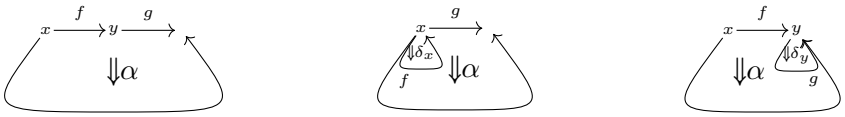
and two variables $\gamma : j \mapsto x \mapsto \varnothing$ and $\gamma' : j' \mapsto z \mapsto \varnothing$. We evaluate the following simple graftings:

$$\begin{aligned}
t(f \leftarrow \gamma) &= \alpha(g \leftarrow \beta)(f \leftarrow \gamma) && \text{well-def. since } \mathtt{ss}\gamma = x = \mathtt{s}f \\
&= \alpha(f \leftarrow \gamma, g \leftarrow \beta) && \text{since } f \in (\mathtt{s}\alpha)^\bullet \\
t(i \leftarrow \gamma') &= \alpha(g \leftarrow \beta)(i \leftarrow \gamma') && \text{well-def. since } \mathtt{ss}\gamma' = z = \mathtt{s}i \\
&= \alpha(g \leftarrow \beta(i \leftarrow \gamma')) && \text{since } i \notin (\mathtt{s}\alpha)^\bullet \\
t(f \leftarrow \gamma)(i \leftarrow \gamma') &= \alpha(f \leftarrow \gamma, g \leftarrow \beta)(i \leftarrow \gamma') && \text{as seen previously} \\
&= \alpha(f \leftarrow \gamma(i \leftarrow \gamma'), g \leftarrow \beta(i \leftarrow \gamma')) && \text{since } i \notin (\mathtt{s}\alpha)^\bullet \\
&= \alpha(f \leftarrow \gamma, g \leftarrow \beta(i \leftarrow \gamma')) && \text{since } i \notin (\mathtt{s}\gamma)^\bullet.
\end{aligned}$$

They can respectively be represented as:



Example 3.28. Consider variables $\alpha : g(y \leftarrow f) \mapsto x \mapsto \varnothing$, $\delta_x : \underline{x} \mapsto x \mapsto \varnothing$, and $\delta_y : \underline{y} \mapsto y \mapsto \varnothing$. Then α, $\alpha(f \leftarrow \delta_x)$, and $\alpha(g \leftarrow \delta_y)$ can respectively be represented as:



Then the sources $\alpha(f \leftarrow \delta_x)$ and $\alpha(g \leftarrow \delta_y)$ are respectively

$$\begin{aligned}
g(y \leftarrow f)[\underline{x}/f] &= g && \text{by } \textbf{(Subst2c)}, \\
g(y \leftarrow f)[\underline{y}/g] &= f && \text{by } \textbf{(Subst2b)},
\end{aligned}$$

and in the first case, the equation $x = y$ is added to the ambient equational theory.

Remark 3.29. The `degen` rule may be replaced by the following `degen-shift` rule without changing the set of derivable sequents of the form $(E \triangleright \Gamma \vdash y : T)$ with $y \in \mathbb{V}$: if $x \in \mathbb{V}_n$ and $d \in \mathbb{V}_{n+2}$ such that $d \notin \mathbb{V}_{\Gamma,n+2}$, then

$$\frac{E \triangleright \Gamma \vdash_n x : T}{E \triangleright \Gamma, d : \underline{x} \mapsto x \mapsto T \vdash_{n+2} d : \underline{x} \mapsto x \mapsto T}\ \mathtt{degen\text{-}shift}$$

However, note that sequents of the form $\left(E \triangleright \Gamma \vdash \underline{y} : T\right)$ are then no longer derivable.

3.1.3 Properties of derivable sequents Let $(E \triangleright \Gamma \vdash x : X)$ be a derivable sequent. We prove in theorem 3.31 that the type $X = (\mathtt{s}x \mapsto \mathtt{ss}x \mapsto \cdots)$ is completely determined by $\mathtt{s}x$ and Γ.

Definition 3.30. We extend the source map $\mathtt{s} : \mathbb{V}_{\Gamma,n} \longrightarrow \mathbb{T}_{\Gamma,n-1}$ to a map $\bar{\mathtt{s}} : \mathbb{T}_{\Gamma,n} \longrightarrow \mathbb{T}_{\Gamma,n-1}$ as follows:

$$\begin{aligned}
\bar{\mathtt{s}} : \mathbb{T}_{\Gamma,n} &\longrightarrow \mathbb{T}_{\Gamma,n-1} \\
x &\longmapsto \mathtt{s}x && x \in \mathbb{V}_\Gamma, \\
\underline{x} &\longmapsto x && x \in \mathbb{V}_\Gamma, \\
x(\overrightarrow{y_i \leftarrow u_i}) &\longmapsto (\mathtt{s}x)[\overrightarrow{\bar{\mathtt{s}}\, u_i / y_i}] && x, \overrightarrow{y_i} \in \mathbb{V}_\Gamma, \overrightarrow{u_i} \in \mathbb{T}_\Gamma.
\end{aligned}$$

Note that by definition, it agrees with $\mathtt{s}$ on variables.

Theorem 3.31. *Let* $(E \triangleright \Gamma \vdash t : s_1 \longmapsto s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing)$ *be a derivable sequent. Then for* $1 \leq k \leq n$ *we have* $s_k = \bar{\mathtt{s}}^k\, t$. *Equivalently, for* $0 \leq i \leq n$, *we have* $\bar{\mathtt{s}}\, s_i = s_{i+1}$, *where by convention,* $s_0 := t$.

Proof. We proceed by induction on the proof tree of the sequent. For readability, we omit equational theories and contexts.

(1) If the sequent is obtained by the following proof tree:

$$\frac{}{x : \varnothing \vdash x : \varnothing}\ \mathtt{point}$$

then $\bar{\mathtt{s}}\, x = \mathtt{s}\, x = \varnothing$, since $x \in \mathbb{V}$, and the result trivially holds.

(2) If the last inference of the proof tree is the following instance of `degen`

$$\frac{\cdots \vdash s_1 : s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing}{\cdots \vdash t : s_1 \longmapsto s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing}\ \mathtt{degen}$$

then $s_1 \in \mathbb{V}$ and $t = \underline{s_1}$. Thus, $\bar{\mathtt{s}}\, t = s_1$, while for $1 \leq i \leq n$, the equality $\bar{\mathtt{s}}\, s_i = s_{i+1}$ holds by induction.

(3) If the last inference of the proof tree is the following instance of `shift`

$$\frac{\cdots \vdash s_1 : s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing}{\cdots \vdash t : s_1 \longmapsto s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing}\ \mathtt{shift}$$

then $t \in \mathbb{V}$, so $\bar{\mathtt{s}}\, t = \mathtt{s}\, t = s_1$, while for $1 \leq i \leq n$, the equality $\bar{\mathtt{s}}\, s_i = s_{i+1}$ holds by induction.

(4) Assume now that the last inference of the proof tree is the following instance of `graft`:

$$\frac{\cdots \vdash u : r_1 \longmapsto r_2 \longmapsto s_3 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing \qquad \cdots \vdash x : X}{\cdots \vdash t : s_1 \longmapsto s_2 \longmapsto \cdots \longmapsto s_n \longmapsto \varnothing}\ \mathtt{graft}\text{-}a$$

with $a \in r_1^\bullet$ and $x \in \mathbb{V}$ such that $\mathtt{ss}\, x = \mathtt{s}\, a$. Then $t = u(a \leftarrow x)$, $s_1 = r_1[\mathtt{s}\, x/a]$, and $s_i = r_i$ for $2 \leq i \leq n$. On the one hand, we have

$$\begin{aligned}
\bar{\mathtt{s}}\, t &= \bar{\mathtt{s}}\,(u(a \leftarrow x)) \\
&= (\bar{\mathtt{s}}\, u)[\bar{\mathtt{s}}\, x/a] \\
&= (\bar{\mathtt{s}}\, u)[\mathtt{s}\, x/a] && \text{since } x \in \mathbb{V} \\
&= r_1[\mathtt{s}\, x/a] && \text{by induction} \\
&= s_1 && \text{by definition.}
\end{aligned}$$

On the other hand, write $r_1 = v(y \leftarrow a(\overrightarrow{z_i \leftarrow w_i}))$, for some $v, \overrightarrow{w_i} \in \mathbb{T}_{n-1}$ and $y \in \mathbb{V}_{n-2}$. This decomposition exhibits r_1 as a grafting (in the sense of definition 3.18) of a term $a(\cdots)$ whose head variable is a onto some term v. We then compute:

$$\begin{aligned}
\bar{\mathtt{s}}\, s_1 &= \bar{\mathtt{s}}\,(r_1[\mathtt{s}\, x/a]) \\
&= \bar{\mathtt{s}}\Big(\ \ v(y \leftarrow a(\overrightarrow{z_i \leftarrow w_i}))\ \ [\mathtt{s}\, x/a]\ \ \Big) \\
&= \bar{\mathtt{s}}\Big(\ \ v\big(y \leftarrow (\mathtt{s}\, x)(\overrightarrow{z_i \leftarrow w_i})\big)\ \ \Big) \\
&= (\bar{\mathtt{s}}\, v)\Big[\ \ (\bar{\mathtt{s}}\mathtt{s}\, x)[\overrightarrow{\bar{\mathtt{s}}\, w_i/z_i}]\ \ /y\Big] && \text{by definition of } \bar{\mathtt{s}} \\
&= (\bar{\mathtt{s}}\, v)\Big[\ \ (\mathtt{ss}\, x)[\overrightarrow{\bar{\mathtt{s}}\, w_i/z_i}]\ \ /y\Big] && \text{by induction} \\
&= (\bar{\mathtt{s}}\, v)\Big[\ \ (\mathtt{s}\, a)[\overrightarrow{\bar{\mathtt{s}}\, w_i/z_i}]\ \ /y\Big] && \text{hypothesis of } \mathtt{graft}\text{-}a \\
&= \bar{\mathtt{s}}\, r_1 && \text{recall } r_1 = v(y \leftarrow a(\overrightarrow{z_i \leftarrow w_i})) \\
&= r_2 = s_2.
\end{aligned}$$

Finally, for $1 \leq i \leq n$, the equality $\bar{\mathtt{s}}\, s_i = s_{i+1}$ holds by induction. □

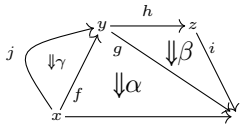
Corollary 3.32. *Let $(E \triangleright \Gamma \vdash t : T)$ be a derivable sequent, and $x : s_1 \mapsto s_2 \mapsto \cdots \mapsto s_n \mapsto \varnothing$ be a typing in Γ. Then for $1 \leq k \leq n$ we have $s_k = \bar{\mathsf{s}}^k t$, or equivalently, for $0 \leq i \leq n$, we have $\bar{\mathsf{s}}\, s_i = s_{i+1}$, with $s_0 := x$.*

Proof. If $x : s_1 \mapsto s_2 \mapsto \cdots \mapsto s_n \mapsto \varnothing$ is a typing in Γ, then somewhere in the proof tree of $(E \triangleright \Gamma \vdash t : T)$ appears a sequent of the form $(F \triangleright \Upsilon \vdash x : s_1 \mapsto s_2 \mapsto \cdots \mapsto s_n \mapsto \varnothing)$, which is necessarily derivable. We conclude by applying theorem 3.31 □

Remark 3.33. A consequence of theorem 3.31 and corollary 3.32 is that at any stage, a context Γ may be replaced by its "meager form" $\bar{\Gamma}$, obtained by replacing "full typings" $y : Y$ by $y : \mathsf{s}y$, i.e. by removing all but the top term of the type Y. Using meager context comes with a cost however: checking the hypothesis of rule `graft` requires to compute the second source $\mathsf{ss}x$ of x, which is not contained in $\bar{\Gamma}$. For clarity, we shall not make use of meager forms throughout the rest of this work.

Convention 3.34. By definition, $\bar{\mathsf{s}}$ extends s to a function $\mathbb{T}_\Gamma \longrightarrow \mathbb{T}_\Gamma$, and for convenience, we just write it as s in the sequel, and call it the *source* of a term.

Example 3.35. Consider the term on the right, representing the pasting diagram on the left:



$$\rightsquigarrow \qquad \alpha(f \leftarrow \gamma, g \leftarrow \beta)$$

Then its source is computed as follows:

$$\begin{aligned}
\mathsf{s}(\alpha(f \leftarrow \gamma, g \leftarrow \beta)) &= (\mathsf{s}\alpha)[(\mathsf{s}\gamma)/f, (\mathsf{s}\beta)/g] && \text{by definition} \\
&= (g(y \leftarrow f))[(\mathsf{s}\gamma)/f, (\mathsf{s}\beta)/g] && \text{since } \mathsf{s}\alpha = g(y \leftarrow f) \\
&= (g(y \leftarrow f))[j/f, (\mathsf{s}\beta)/g] && \text{since } \mathsf{s}\gamma = j \\
&= (g(y \leftarrow f))[j/f, i(z \leftarrow h)/g] && \text{since } \mathsf{s}\beta = i(z \leftarrow h) \\
&= (g(y \leftarrow j))[i(z \leftarrow h)/g] && \text{see equation (3.22)} \\
&= (i(z \leftarrow h))(y \leftarrow j) && \text{see equation (3.22)} \\
&= i(z \leftarrow h(y \leftarrow j)) && \text{since } y \in (\mathsf{s}h)^\bullet.
\end{aligned}$$

The latter term indeed corresponds to the source of the pasting diagram, which is the arrow composition on the top.

Lemma 3.36 (Unique occurence lemma)**.** *Let $(E \triangleright \Gamma \vdash_n t : s_1 \mapsto \cdots)$ be a derivable sequent, where $t \in \mathbb{T}_n$ is not degenerate, say $t = x(\overrightarrow{a_i \leftarrow u_i})$.*

(1) Let $y \in t^\bullet$. Then either $y =_E x$, or $y \in u_i^\bullet$ for a unique i.

(2) Let $b \in t^!$. Then either $b \in x^! - \{a_1, \ldots\}$, or $b \in u_i^!$ for a unique i.

Proof. (1) By assumption of the `graft` rule, each n-variable of t occurs only once in t.

(2) By the first point, each $(n-1)$-variable of s_1 occurs exactly once. By theorem 3.31, $s_1 = (\mathsf{s}x)[\overrightarrow{\bar{\mathsf{s}}u_i/a_i}]$. Thus b either occurs in $\mathsf{s}x$ or on $\mathsf{s}u_i$ for a unique i. □

Proposition 3.37. *Let $(E \triangleright \Gamma \vdash t : T)$ be a derivable sequent, where $t \in \mathbb{T}_\Gamma$.*

(1) Let $a \in \mathbb{V}_\Gamma$ be a variable of type A. Then the sequent $(E|_a \triangleright \Gamma|_a \vdash a : A)$ is derivable, where $E|_a$ (resp. $\Gamma|_a$) is the restriction of E (resp. Γ) to a and variables occurring in A.

(2) If u is a subterm of t, then the restricted sequent $(E|_u \triangleright \Gamma|_u \vdash u : U)$ is derivable, where $U = \mathtt{s}\,u \mapsto \mathtt{s}^2\,u \mapsto \cdots$, and where $E|_u$ (resp. $\Gamma|_u$) is the restriction of E (resp. Γ) to variables occurring in u and U.

Proof. (1) (a) If a is 0-dimensional, then $(E|_a \triangleright \Gamma|_a \vdash a : A) = (\triangleright a : \varnothing \vdash a : \varnothing)$ can be obtained by an instance of rule `point`.

(b) If $a = x$, then $(E|_a \triangleright \Gamma|_a \vdash a : A) = (E \triangleright \Gamma \vdash x : X)$ is derivable by assumption.

(c) Otherwise, a first appears in the conclusion of an instance of `shift` in the proof tree of $(E \triangleright \Gamma \vdash x : X)$. Then $(E|_a \triangleright \Gamma|_a \vdash a : A)$ is the conclusion of that instance, and is derivable.

(2) (a) If $u = t$, then the claim trivially holds.

(b) If $t = x(\overrightarrow{y_i \leftarrow v_i})$, $1 \le i \le k$ and $u = v_j$ for some j, then on the bottom of the proof tree of $(E \triangleright \Gamma \vdash t : T)$ is a sequence of k instances of the `graft` rule, and $(E|_u \triangleright \Gamma|_u \vdash u : U)$ was the right premise of one of them.

(c) If $t = x(\overrightarrow{y_i \leftarrow v_i})$, and u is a subterm of v_j for some j, then by the previous point, $(E|_{v_j} \triangleright \Gamma|_{v_j} \vdash v_j : V_j)$ is derivable, and inductively, $((E|_{v_j})_u \triangleright (\Gamma|_{v_j})_u \vdash U : U) = (E|_u \triangleright \Gamma|_u \vdash u : U)$ is derivable. □

3.2 Equivalence with polynomial opetopes In this section, all sequents are assumed derivable in Opt$^!$. We show that sequents typing a variable (up to α-equivalence) are in bijective correspondence with the "polynomial" opetopes of definition 2.35. To this end, we define the *polynomial coding* operation $[\![-]\!]_{n+1}$ that maps a sequent $(E \triangleright \Gamma \vdash_n t : T)$ typing an n-term $t \in \mathbb{T}_n$, to an $(n+1)$-opetope $[\![E \triangleright \Gamma \vdash_n t : T]\!]_{n+1} \in \mathbb{O}_{n+1}$, written $[\![t : T]\!]_{n+1}$ or even $[\![t]\!]_{n+1}$ for short, if no ambiguity arises.

The idea of the polynomial coding is to map a pasting diagram described by a term (on the left) to its underlying composition tree, and reapply the coding recursively (on the right):

$$[\![\alpha(g \leftarrow \beta)]\!] \quad = \quad \left[\!\!\left[\begin{array}{ccc} & h & \quad i \\ f & & \beta \\ & \alpha & \quad g \end{array} \right]\!\!\right] \quad := \quad \begin{array}{ccc} & [\![h]\!] & \quad [\![i]\!] \\ [\![f]\!] & & [\![\beta]\!] \\ & [\![\alpha]\!] & \quad [\![g]\!] \end{array}$$

For $t \in \mathbb{T}_n$ and $z \in t^\bullet$, the *address* $\&_t\, z \in \mathbb{A}_n$ of z in t is an n-address (see section 2.2.3) that, much like in trees (definition 2.8), indicates "where z is located in t":

Definition 3.38 (Address in a term)**.** Let $(E \triangleright \Gamma \vdash \cdots)$ be a derivable sequent, $t \in \mathbb{T}_\Gamma$ be a non-degenerate typed term, say $t = x(\overrightarrow{y_i \leftarrow u_i})$. From proposition 3.37, we can extract a typing for each of the u_is from the proof tree of $(E \triangleright \Gamma \vdash \cdots)$. Assume further that for all subterms $z(a_j \leftarrow v_j)$ that occur in t, we have $a_j \in (\mathtt{s}\,z)^\bullet$, for all j.

(1) Let $z \in t^\bullet$. By lemma 3.36, either $z =_E x$, or $z \in u_i^\bullet$ for a unique i. The *address* $\&_t\, z \in \mathbb{A}_n$ of z in t is defined as

$$\&_t\, z := \begin{cases} [] & \text{if } z =_E x, \\ [\&_{\mathtt{s}\,x}\, y_i] \cdot \&_{u_i}\, z & \text{if } z \in u_i^\bullet. \end{cases}$$

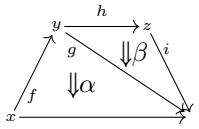
If $[p] = \&_t\, z$, then we write $\mathsf{v}_{[p]}\, t := z$ for the variable of t at address $[p]$. In particular, $\mathsf{v}_{[]}\, t = x$.

(2) Let $a \in t^!$. By lemma 3.36, either $a \in x^! - \{y_1, \ldots\}$, or $a \in u_i^!$ for a unique i. The *address* $\&_t a \in \mathbb{A}_n$ of a in t is defined as

$$\&_t a := \begin{cases} [\&_{\mathbf{s}\, x}\, a] & \text{if } a \in x^!, \\ [\&_{\mathbf{s}\, x}\, y_i] \cdot \&_{u_i}\, a & \text{if } a \in u_i^!. \end{cases}$$

Of course, those addresses tacitly depend on the surrounding sequent.

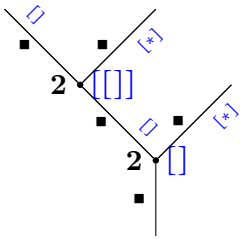
Example 3.39. The context describing the pasting diagram:



contains the following typings: $x, y, z : \varnothing$, $f : x \mapsto \varnothing$, $g : y \mapsto \varnothing$, $h : y \mapsto \varnothing$, $i : z \mapsto \varnothing$, $\alpha : g(y \leftarrow f) \mapsto a \mapsto \varnothing$, and $\beta : i(z \leftarrow h) \mapsto b \mapsto \varnothing$. Write $t := \alpha(g \leftarrow \beta)$. Then,

$$\begin{aligned} \&_t\, \alpha &= [], \\ \&_t\, \beta &= [\&_{\mathbf{s}\, \alpha}\, g] \cdot \&_\beta\, \beta = [[]] \cdot [] = [[]], \\ \&_t\, i &= [\&_{\mathbf{s}\, \alpha}\, g] \cdot \&_\beta\, i = [[]] \cdot [\&_{\mathbf{s}\, \beta}\, i] = [[]] \cdot [[]] = [[][]], \\ \&_t\, h &= [\&_{\mathbf{s}\, \alpha}\, g] \cdot \&_\beta\, h = [[]] \cdot [\&_{\mathbf{s}\, \beta}\, h] = [[]] \cdot [[\&_{\mathbf{s}\, i}\, z] \cdot \&_h\, h] = [[][[]]] = [[][*]], \\ \&_t\, f &= [\&_{\mathbf{s}\, \alpha}\, f] = [[\&_{\mathbf{s}\, g}\, y] \cdot \&_f\, f] = [[*]]. \end{aligned}$$

Those addresses indeed match with those of the (intuitively) corresponding opetope:



We now present the *polynomial coding operation* $[\![-]\!]$, mapping the derivable sequents of system Opt$^!$ to opetopes. In theorem 3.64, we prove that it is a bijection. Unsurprisingly, $[\![-]\!]$ needs to be defined by induction on the dimension of the term being typed by the sequent. However, as we shall see, a secondary induction is also needed, this time on the number of variables in the term. In the sequel, *between definition 3.40 and theorem 3.64 (included), every definition and result is stated assuming that theorem 3.64 holds by induction.*

Definition 3.40 (Polynomial coding for variables)**.** The polynomial coding operation $[\![-]\!]_n$ is defined on derivable sequents typing n-variables as follows:

$$[\![E \triangleright \Gamma \vdash_0 x : \varnothing]\!]_0 := \blacklozenge, \tag{3.41}$$

$$[\![E \triangleright \Gamma \vdash_1 x : a \mapsto \varnothing]\!]_1 := \blacksquare, \tag{3.42}$$

$$[\![E \triangleright \Gamma, x : t \mapsto T \vdash_n x : t \mapsto T]\!]_n := [\![E \triangleright \Gamma \vdash_{n-1} t : T]\!]_n, \tag{3.43}$$

where in the case (3.43), the value of $[\![t]\!]_{n-1}$ is given in definition 3.44.

Definition 3.44 (Polynomial coding)**.** The polynomial coding operation $[\![-]\!]_{n+1}$ is defined on derivable sequents typing an n-term inductively as follows:

$$[\![E \triangleright \Gamma \vdash_{n+1} \underline{x} : x \longmapsto \cdots]\!]_{n+2} := \mathsf{I}_{[\![x]\!]_n}, \tag{3.45}$$

$$[\![E \triangleright \Gamma \vdash_n x(\overrightarrow{y_i \leftarrow u_i}) : \cdots]\!]_{n+1} := \mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathsf{s}x}\, y_i]} [\![u_i]\!]_{n+1}\,. \tag{3.46}$$

By proposition 3.37, in equation (3.46), we can extract a typing for each of the u_is from the proof tree of $\big(E \triangleright \Gamma \vdash_n x(\overrightarrow{y_i \leftarrow u_i}) : \cdots\big)$. Further, if $x \in \mathbb{V}_n$ is seen as an n-term, then $[\![x]\!]_{n+1} = \mathsf{Y}_{[\![x]\!]_n}$. If $(E \triangleright \Gamma \vdash \cdots)$ is a derivable sequent, and $a \in \mathbb{V}_\Gamma$, then the restricted sequent $(E|_a \triangleright \Gamma|_a \vdash a : A)$ is also derivable by proposition 3.37, and we may construct an opetope $[\![a]\!] = [\![E|_a \triangleright \Gamma|_a \vdash a : A]\!]$.

Note that definitions 3.40 and 3.44 are mutually dependent. The coding function $[\![-]\!]_n$ can be considered in two ways: first as mapping derivable sequents typing n-variables to n-opetopes, for any n (definition 3.40), or as mapping derivable sequents typing an $(n-1)$-term to an n-opetope, for $n \geq 2$ (definition 3.44). The former is more faithful to the geometrical intuition that an n-opetope is first and foremost an n-dimensional shape, in this case encoded by an n-variable (see also example 3.8). However it is the latter version that we need to study in details. For example, it is clear that the coding function is well-defined in cases (3.41), (3.42), and (3.45), and if $n = 1$, then case (3.46) is well-defined too, as there is a unique 1-opetope. But in order to establish that case (3.46) is always well-defined, we need to inductively rely on properties of $[\![-]\!]$ proved in the sequel.

Lemma 3.47. *Let $(E \triangleright \Gamma \vdash_n t : T)$ be a sequent such that $[\![t]\!]_{n+1}$ is well-defined. Then $[\![t]\!]^\bullet_{n+1} = \{\&_t\, x \mid x \in t^\bullet\}$, i.e. the node addresses of the opetope $[\![t]\!]_{n+1}$ are exactly the addresses of the n-variables of t.*

Proof. If $n = 0$, or if t is degenerate, then the result trivially holds. If $t = x(\overrightarrow{y_i \leftarrow u_i})$ as in equation (3.46), we have

$$\begin{aligned}
[\![t]\!]^\bullet_{n+1} &= [\![x(\overrightarrow{y_i \leftarrow u_i})]\!]^\bullet_{n+1} \\
&= \left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathsf{s}x}\, y_i]} [\![u_i]\!]_{n+1}\right)^\bullet && \text{by equation (3.46)} \\
&= \{[]\} \bigcup_i \{[\&_{\mathsf{s}x}\, y_i] \cdot [p] \mid [p] \in ([\![u_i]\!]_{n+1})^\bullet\} \\
&= \{[]\} \bigcup_i \{[\&_{\mathsf{s}x}\, y_i] \cdot \&_t\, x \mid x \in u_i^\bullet\} && \text{by induction} \\
&= \{\&_t\, x \mid x \in t^\bullet\} && \text{see definition 3.38.}
\end{aligned}$$

□

Proposition 3.48. *Let $(E \triangleright \Gamma \vdash_n t : T)$ be a derivable sequent where t is not degenerate, say $t = x(\overrightarrow{y_i \leftarrow u_i})$. Assume that $[\![t]\!]_{n+1}$ is well-defined. For $[p] \in [\![t]\!]^\bullet_{n+1}$ and $z := \mathsf{v}_{[p]}\, t$ we have $\mathsf{s}_{[p]}\, [\![t]\!]_{n+1} = [\![z]\!]_n$.*

Proof. If $n = 0$, then t is necessarily a 0-variable, so the only possible address in t is $[]$. Then, $\mathsf{s}_{[]}\, [\![t]\!]_1 = \mathsf{s}_{[]}\, \blacksquare = \blacklozenge = [\![t]\!]_0 = [\![\mathsf{v}_{[]}\, t]\!]_0$. Assume that $n \geq 1$. By definition,

$$[\![t]\!]_{n+1} = \mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathsf{s}x}\, y_i]} [\![u_i]\!]_{n+1}\,,$$

and we distinguish two cases. If $z = x$, then $[p] = []$, and the result clearly holds. Otherwise, $[p] = [\&_{\mathsf{s}\,x}\, y_j] \cdot \&_{u_j}\, z$, where j is the unique index such that $z \in u_j^\bullet$ (see lemma 3.36). Then,

$$\begin{aligned}
\mathsf{s}_{[p]}\, [\![t]\!]_{n+1} &= \mathsf{s}_{[\&_{\mathsf{s}\,x}\, y_j]\cdot\&_{u_i}\, z} \left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathsf{s}\,x}\, y_i]} [\![u_i]\!]_{n+1} \right) \\
&= \mathsf{s}_{\&_{u_j}\, z}\, [\![u_j]\!]_{n+1} \\
&= [\![z]\!]_n && \text{by induction.}
\end{aligned}$$

□

Corollary 3.49. *Let x and y_i be as in equation (3.46), and assume that $[\![x]\!]_n$ and $[\![y_i]\!]_{n-1}$ are both well-defined. Then for $[p_i] := \&_{\mathsf{s}\,x}\, y_i$ we have $\mathsf{s}_{[p_i]}\, [\![x]\!]_n = [\![y_i]\!]_{n-1}$.*

Proof. We simply have

$$\begin{aligned}
\mathsf{s}_{[p_i]}\, [\![x]\!]_n &= \mathsf{s}_{[p_i]}\, [\![\mathsf{s}\, x]\!]_n && \text{see equation (3.43)} \\
&= [\![y_i]\!]_{n-1} && \text{by proposition 3.48.}
\end{aligned}$$

□

Lemma 3.50. *Let $n \geq 1$, and consider the following instance of the* `graft` *rule:*

$$\frac{\cdots \vdash_n t : T \qquad \cdots \vdash_n x : X}{\cdots \vdash_n t(a \leftarrow x) : R}\ \texttt{graft-}a$$

where $(\cdots \vdash_n t : T)$ and $(\cdots \vdash_n x : X)$ are derivable. Writing $r := t(a \leftarrow x)$, and assuming that $[\![r]\!]_{n+1}$ is well-defined, we have $[\![r]\!]_{n+1} = [\![t]\!]_{n+1} \circ_{\&_t\, a} \mathsf{Y}_{[\![x]\!]_n}$.

Proof. By equation (3.46), $[\![r]\!]_{n+1} = [\![t]\!]_{n+1} \circ_{[l]} \mathsf{Y}_{[\![x]\!]_n}$, for some leaf address $[l]$. By assumption of the `graft` rule, t is non-degenerate, and write it as $t = z(\overrightarrow{y_i \leftarrow u_i})$. According to definition 3.18,

$$r = z(\overrightarrow{y_i \leftarrow u_i})(a \leftarrow x) = \begin{cases} z(\overrightarrow{y_i \leftarrow u_i}, a \leftarrow x) & \text{if } a \in (\mathsf{s}\, z)^\bullet, \\ z(\overrightarrow{y_i \leftarrow u_i(a \leftarrow x)}) & \text{if } a \notin (\mathsf{s}\, z)^\bullet, \end{cases}$$

so there are two cases.

(1) If $a \in (\mathsf{s}\, z)^\bullet$, then $[l] = [\&_{\mathsf{s}\, z}\, a] = \&_t\, a$.

(2) If $a \notin (\mathsf{s}\, z)^\bullet$, then

$$[\![r]\!]_{n+1} = \mathsf{Y}_{[\![z]\!]_n} \bigcirc_{[\&_{\mathsf{s}\, z}\, y_i]} [\![u_i(a \leftarrow x)]\!]_{n+1}\,.$$

Let j be the unique index such that $a \in (\mathsf{s}\, u_j)^\bullet$. By induction,

$$[\![u_j(a \leftarrow x)]\!]_{n+1} = [\![u_j]\!]_{n+1} \circ_{\&_{u_j}\, a} \mathsf{Y}_{[\![x]\!]_n},$$

thus $[l] = [\&_{\mathsf{s}\, z}\, y_j] \cdot \&_{u_j}\, a = \&_t\, a$. □

Lemma 3.51 (Named readdressing lemma)**.** *Let $n \geq 1$ and $(E \triangleright \Gamma \vdash_n r : R)$ be a derivable sequent such that $[\![r]\!]_{n+1}$ is well-defined. For $b \in (\mathsf{s}\, r)^\bullet$, we have $\&_{\mathsf{s}\, r}\, b = \wp_{[\![r]\!]_{n+1}}\, \&_r\, b$ (recall the readdressing map $\wp$ from definition 2.27).*

Proof. If r is a variable, then by equation (3.46), $[\![r]\!]_{n+1} = \mathsf{Y}_{[\![\mathtt{s}r]\!]_n}$, and

$$\begin{aligned} \&_{\mathtt{s}r}\, b &= \wp_{\mathsf{Y}_{[\![r]\!]_n}}[\&_{\mathtt{s}r}\, b] && \text{by the results of appendix A.1} \\ &= \wp_{[\![r]\!]_{n+1}}\, \&_r\, b && \text{see definition 3.38.} \end{aligned}$$

If r is degenerate, say $r = \underline{x}$, then the result trivially holds as $\mathtt{s}r = x$ only has one variable address. Otherwise, the sequent follows from an instance of the `graft` rule, say

$$\frac{\cdots \vdash_n t : T \qquad \cdots \vdash_n x : X}{\cdots \vdash_n r : R}\ \mathtt{graft}\text{-}a$$

where by induction, $\&_{\mathtt{s}t}\, a = \wp_{[\![t]\!]_{n+1}}\, \&_t\, a$ for all $a \in (\mathtt{s}t)^\bullet$. Since $\mathtt{s}r = (\mathtt{s}t)[\mathtt{s}x/a]$, we have

$$\begin{aligned} \&_{\mathtt{s}r}\, b &= \begin{cases} \&_{\mathtt{s}t}\, a \cdot \&_{\mathtt{s}x}\, b & \text{if } b \in (\mathtt{s}x)^\bullet, \\ \&_{\mathtt{s}t}\, a \cdot \&_{\mathtt{s}x}\, c \cdot [p] & \text{if } b \in (\mathtt{s}t)^\bullet, \&_{\mathtt{s}t}\, a \sqsubseteq \&_{\mathtt{s}t}\, b, \\ & \quad \text{say } \&_{\mathtt{s}t}\, b = \&_{\mathtt{s}t}\, a \cdot [\&_{\mathtt{s}a}\, c] \cdot [p], \\ \&_{\mathtt{s}t}\, b & \text{if } b \in (\mathtt{s}t)^\bullet, \&_{\mathtt{s}t}\, a \notin \&_{\mathtt{s}t}\, b. \end{cases} \\ &= \wp_{\left([\![t]\!]_{n+1} \circ_{\&_t\, a} \mathsf{Y}_{[\![x]\!]_n}\right)}\, \&_r\, b && \text{see theorem A.14} \\ &= \wp_{[\![r]\!]_{n+1}}\, \&_r\, b && \text{by equation (3.46).} \end{aligned}$$

□

Lemma 3.52. *Let $(E \triangleright \Gamma \vdash \cdots)$ be a derivable sequent, $u, v \in \mathbb{T}_{\Gamma,n}$ be n-terms, where u is non-degenerate, say $u = y(\overrightarrow{a_i \leftarrow w_i})$. Let $x \in u^\bullet$ be a variable such that $\mathtt{s}x = \mathtt{s}v$. In particular, the substitution $u[v/x]$ is well-defined. Further, assume that $\mathtt{t}[\![v]\!]_{n+1} = [\![x]\!]_n$. Then $[\![u[v/x]]\!]_{n+1} = [\![u]\!]_{n+1} \,{}_{\&_u\, x}\!\square\, [\![v]\!]_{n+1}$.*

Proof. (1) If $x =_E y$, then

$$\begin{aligned} & [\![u[v/x]]\!]_{n+1} \\ &= \left[\!\!\left[v(\overrightarrow{a_i \leftarrow w_i})\right]\!\!\right]_{n+1} \\ &= [\![v]\!]_{n+1} \bigcirc_{\&_v\, a_i} [\![w_i]\!]_{n+1} && \spadesuit \\ &= \left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\wp_{[\![v]\!]_{n+1}}\, \&_v\, a_i]} [\![w_i]\!]_{n+1}\right) \,{}_{[]}\!\square\, [\![v]\!]_{n+1} && \text{since } \mathtt{t}[\![v]\!]_{n+1} = [\![x]\!]_n \\ &= \left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathtt{s}v}\, a_i]} [\![w_i]\!]_{n+1}\right) \,{}_{[]}\!\square\, [\![v]\!]_{n+1} && \diamondsuit \\ &= \left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathtt{s}x}\, a_i]} [\![w_i]\!]_{n+1}\right) \,{}_{[]}\!\square\, [\![v]\!]_{n+1} && \text{by assumption} \\ &= [\![u]\!]_{n+1} \,{}_{[]}\!\square\, [\![v]\!]_{n+1} \\ &= [\![u]\!]_{n+1} \,{}_{\&_u\, x}\!\square\, [\![v]\!]_{n+1} && \clubsuit \end{aligned}$$

where ♠ is by equation (3.46), ♢ follows from lemma 3.51, and ♣ follows from the fact that x is the head variable of u (see definition 3.38).

(2) If $x \neq_E y$, then by lemma 3.36, there is a unique index j such that $x \in w_j^\bullet$. We have

$$
\begin{aligned}
&[\![u[v/x]]\!]_{n+1} \\
&= [\![y(\ldots, a_j \leftarrow w_j[v/x], \ldots)]\!]_{n+1} \\
&= \left(\mathsf{Y}_{[\![y]\!]_n} \bigcirc_{[\&_{\mathsf{s}\,y}\, a_i], i\neq j} [\![w_i]\!]_{n+1}\right) \circ_{[\&_{\mathsf{s}\,y}\, a_j]} [\![w_j[v/x]]\!]_{n+1} && \spadesuit \\
&= \left(\mathsf{Y}_{[\![y]\!]_n} \bigcirc_{[\&_{\mathsf{s}\,y}\, a_i], i\neq j} [\![w_i]\!]_{n+1}\right) \\
&\qquad \circ_{[\&_{\mathsf{s}\,y}\, a_j]} \left([\![w_j]\!]_{n+1} \,\square_{\&_{w_j}\, x}\, [\![v]\!]_{n+1}\right) && \text{by induction} \\
&= [\![u]\!]_{n+1} \,\square_{[\&_{\mathsf{s}\,y}\, a_j]\cdot \&_{w_j}\, x}\, [\![v]\!]_{n+1} && \diamondsuit \\
&= [\![u]\!]_{n+1} \,\square_{\&_u\, x}\, [\![v]\!]_{n+1} && \clubsuit
\end{aligned}
$$

where $\spadesuit$ is by equation (3.46), $\diamondsuit$ is just a rearrangement of terms, and $\clubsuit$ is definition 3.38. □

Proposition 3.53. *Let $n \geq 1$ and $(E \triangleright \Gamma \vdash_n r : R)$ be a derivable sequent such that $[\![r]\!]_{n+1}$ is well-defined. Then $\mathsf{t}\,[\![r]\!]_{n+1} = [\![\mathsf{s}\,r]\!]_n$.*

Proof. (1) If $r \in \mathbb{V}_n$ is a variable, then

$$
\begin{aligned}
\mathsf{t}\,[\![r]\!]_{n+1} &= \mathsf{t}\,\mathsf{Y}_{[\![r]\!]_n} && \text{by equation (3.46)} \\
&= [\![r]\!]_n\,.
\end{aligned}
$$

(2) If r is a degenerate term, say $r = \underline{x}$ for $x \in \mathbb{V}_{n-1}$, then

$$
\begin{aligned}
\mathsf{t}\,[\![r]\!]_{n+1} &= \mathsf{t}\,\mathsf{I}_{[\![x]\!]_{n-1}} && \text{by equation (3.45)} \\
&= \mathsf{Y}_{[\![x]\!]_{n-1}} \\
&= [\![x]\!]_n && \text{by equation (3.46)} \\
&= [\![\mathsf{s}\,r]\!]_n && \text{see rule } \texttt{degen}.
\end{aligned}
$$

(3) Otherwise, the sequent follows from an instance of the `graft` rule, say

$$
\frac{\cdots \vdash_n t : T \qquad \cdots \vdash_n x : X}{\cdots \vdash_n r : R}\ \texttt{graft-}a
$$

Let $[p] := \wp_{[\![t]\!]_{n+1}} \&_t\, a$. We have

$$
\begin{aligned}
\mathsf{t}\,[\![r]\!]_{n+1} &= \mathsf{t}\,[\![t(a \leftarrow x)]\!]_{n+1} && \text{by definition of } \texttt{graft-}a \\
&= \mathsf{t}\left([\![t]\!]_{n+1} \circ_{\&_{\mathsf{s}\,t}\, a} \mathsf{Y}_{[\![x]\!]_n}\right) && \text{by lemma 3.50} \\
&= \mathsf{t}\,[\![t]\!]_{n+1} \,\square_{[p]}\, [\![x]\!]_n && \text{by proposition 2.37} \\
&= [\![\mathsf{s}\,t]\!]_n \,\square_{[p]}\, [\![x]\!]_n && \text{by induction} \\
&= [\![\mathsf{s}\,t]\!]_n \,\square_{\&_{\mathsf{s}\,t}\, a}\, [\![x]\!]_n && \text{by lemma 3.51} \\
&= [\![(\mathsf{s}\,t)[x/a]]\!]_n && \text{by lemma 3.52} \\
&= [\![\mathsf{s}\,r]\!]_n && \text{by definition of } \texttt{graft-}a.
\end{aligned}
$$

□

Corollary 3.54. *Let $n \geq 1$ and $(E \triangleright \Gamma \vdash_n t : T)$ be a derivable sequent such that $[\![t]\!]_{n+1}$ is well-defined. Then $\&_t$ exhibits a bijection*

$$(\mathtt{s}\, t)^\bullet \xrightarrow{\cong} [\![t]\!]^!_{n+1}\,.$$

Proof. Since the readdressing map $\wp_{[\![t]\!]_{n+1}}$ is a bijection, $\&_t$ can be expressed as the following composite:

$$\begin{aligned}
&(\mathtt{s}\, t)^\bullet \xrightarrow{\&_t} \{\&_t\, a \mid a \in (\mathtt{s}\, t)^\bullet\} \\
&\xrightarrow{\wp_{[\![t]\!]_{n+1}}} \{\&_{\mathtt{s}\, t}\, a \mid a \in (\mathtt{s}\, t)^\bullet\} && \text{by lemma 3.51} \\
&= [\![\mathtt{s}\, t]\!]^\bullet_n && \text{by lemma 3.47} \\
&= (\mathtt{t}\, [\![t]\!]_{n+1})^\bullet && \text{by proposition 3.53} \\
&\xrightarrow{\wp^{-1}_{[\![t]\!]_{n+1}}} [\![t]\!]^!_{n+1}\,.
\end{aligned}$$

□

Proposition 3.55. *With variables as in equation (3.46), we have that for all i*

$$\mathtt{e}_{[]}\, [\![u_i]\!]_{n+1} \;=\; \mathtt{s}_{\&_{\mathtt{s}\, x}\, y_i}\, [\![x]\!]_n\,,$$

and the graftings are well-defined.

Proof. Write u_i as $a(\overrightarrow{b_j \leftarrow v_j})$, and consider

$$\begin{aligned}
\mathtt{e}_{[]}\, [\![u_i]\!]_{n+1} &= \mathtt{t}\mathtt{s}_{[]}\, [\![u_i]\!]_{n+1} \\
&= \mathtt{t}\, [\![a]\!]_n && \text{by proposition 3.48} \\
&= \mathtt{t}\, [\![\mathtt{s}\, a]\!]_n && \text{see equation (3.43)} \\
&= [\![\mathtt{s}\mathtt{s}\, a]\!]_{n-1} && \text{by proposition 3.53} \\
&= [\![\mathtt{s}\, y_i]\!]_{n-1} && \text{by the conditions of } \mathtt{graft} \\
&= [\![y_i]\!]_{n-1} && \text{see equation (3.43)} \\
&= \mathtt{s}_{\&_{\mathtt{s}\, x}\, y_i}\, [\![x]\!]_n && \text{by corollary 3.49.}
\end{aligned}$$

□

This result concludes the proof that cases (3.41), (3.42), (3.45), and (3.46). The rest of this section is dedicated to proving theorem 3.64 stating that $[\![-]\!]_n$ is a bijection modulo α-equivalence. Recall that we are still in an induction spanning from definition 3.40 to theorem 3.64.

Definition 3.56. We define the *named coding function* $C^!$, that maps an n-opetope to a sequent typing an n-variable, as follows.

(1) Trivially, $C^!\,(\blacklozenge)$ is obtained by the following proof tree:

$$\frac{}{C^!\,(\blacklozenge)}\,\mathtt{point} \tag{3.57}$$

with an arbitrary choice of variable (different choices lead to equivalent sequents).

(2) For $\phi \in \mathbb{O}_{n-2}$ the sequent $C^!\left(\mathsf{I}_\phi\right)$ is obtained by the following proof tree:

$$\dfrac{\dfrac{\vdots}{C^!\left(\phi\right) = \left(E \triangleright \Gamma \vdash_{n-2} x : X\right)}\,\mathtt{degen}}{\dfrac{E \triangleright \Gamma \vdash_{n-1} \underline{x} : x \longmapsto X}{C^!\left(\mathsf{I}_\phi\right)}\,\mathtt{shift}} \tag{3.58}$$

(3) For $\psi \in \mathbb{O}_{n-1}$, the sequent $C^!\left(\mathsf{Y}_\psi\right)$ is obtained by the following proof tree:

$$\dfrac{\dfrac{\vdots}{C^!\left(\psi\right)}}{C^!\left(\mathsf{Y}_\psi\right)}\,\mathtt{shift} \tag{3.59}$$

with an arbitrary choice of fresh variable (different choices lead to equivalent sequents).

(4) Let $n \geq 2$, $\nu \in \mathbb{O}_n$ having at least one node, $[l] \in \nu^!$, and $\psi \in \mathbb{O}_{n-1}$ be such that the grafting $\nu \circ_{[l]} \mathsf{Y}_\psi$ is well-defined. By induction, the proof trees of $C^!\left(\nu\right)$ and $C^!\left(\psi\right)$ both end with an instance of rule `shift`, and write the former as

$$\dfrac{\dfrac{\vdots}{E \triangleright \Gamma \vdash_{n-1} t : T}}{C^!\left(\nu\right)}\,\mathtt{shift}$$

Then the sequent $C^!\left(\nu \circ_{[l]} \mathsf{Y}_\psi\right)$ is obtained by the following proof tree:

$$\dfrac{\dfrac{\begin{array}{cc}\vdots & \vdots \\ E \triangleright \Gamma \vdash_{n-1} t : T & C^!\left(\psi\right)\end{array}}{F \triangleright \Upsilon \vdash_{n-1} u : U}\,\mathtt{graft}\text{-}a}{C^!\left(\nu \circ_{[l]} \mathsf{Y}_\psi\right)}\,\mathtt{shift} \tag{3.60}$$

where the variable $a \in (\mathtt{s}t)^\bullet$ is an $(n-2)$-variable such that $\&_{\mathtt{s}t}\, a = \wp_\nu[l]$ (see corollary 3.54), and where the adequate α-conversions have been performed to fulfill the side conditions of `graft`.

Proposition 3.61. *In proof tree* (3.60)*, the instance of* `graft` *is well-defined.*

Proof. If $n = 2$, then all graftings are well-defined, as there exists only one 1-opetope. Assume that $n > 2$, write $C^!\left(\psi\right) = \left(\cdots \vdash_{n-1} p : P\right)$, where $p \in \mathbb{V}_n$, and let $[q] := \wp_\nu[l]$. By induction on the number of nodes, and by theorem 3.64, we have that $[\![t]\!]_n = \left[\!\left[C^!\left(\nu\right)\right]\!\right]_n = \nu$. We have

$$\begin{aligned}
[\![\mathtt{s}\,a]\!]_{n-2} &= [\![a]\!]_{n-2} && \text{see equation (3.43)}\\
&= \left[\!\left[\mathtt{v}_{[q]}\,\mathtt{s}\,t\right]\!\right]_{n-2} && \text{by definition of } a\\
&= \mathtt{s}_{[q]}\,[\![\mathtt{s}\,t]\!]_{n-1} && \text{by proposition 3.48}\\
&= \mathtt{s}_{[q]}\,\mathtt{t}\,[\![t]\!]_n && \text{by proposition 3.53}\\
&= \mathtt{s}_{[q]}\,\mathtt{t}\,\nu && \text{by inductively applying theorem 3.64}\\
&= \mathtt{e}_{[l]}\,\nu && \text{by } \textbf{(Glob2)}\\
&= \mathtt{t}\,\psi && \text{by assumption}\\
&= \mathtt{t}\,[\![p]\!]_{n-1} && \text{by definition}\\
&= \mathtt{t}\,[\![\mathtt{s}\,p]\!]_{n-1} && \text{see equation (3.43)}\\
&= [\![\mathtt{s}\,\mathtt{s}\,p]\!]_{n-2} && \text{by proposition 3.53.}
\end{aligned}$$

By applying theorem 3.64 inductively, the polynomial coding $[\![-]\!]_{n-2}$ is injective modulo α-equivalence. Hence without loss of generality, we can assume $\mathtt{s}\,a = \mathtt{s}\,\mathtt{s}\,p$, and finally, the instance of the `graft` rule is well-defined. □

Proposition 3.62. *Let $n \geq 2$ and $\omega \in \mathbb{O}_n$ have at least three nodes. The sequent $C^!(\omega)$ does not depend on the decomposition of ω in corollas. Explicitly, for any two decompositions of ω, say*

$$\begin{aligned}\omega &= \left(\cdots \left(\mathsf{Y}_{\mathtt{s}_{[p_1]}\omega} \underset{[p_2]}{\circ} \mathsf{Y}_{\mathtt{s}_{[p_2]}\omega} \right) \underset{[p_3]}{\circ} \mathsf{Y}_{\mathtt{s}_{[p_3]}\omega} \cdots \right) \underset{[p_k]}{\circ} \mathsf{Y}_{\mathtt{s}_{[p_k]}\omega} \\ &= \left(\cdots \left(\mathsf{Y}_{\mathtt{s}_{[q_1]}\omega} \underset{[q_2]}{\circ} \mathsf{Y}_{\mathtt{s}_{[q_2]}\omega} \right) \underset{[q_3]}{\circ} \mathsf{Y}_{\mathtt{s}_{[q_3]}\omega} \cdots \right) \underset{[q_k]}{\circ} \mathsf{Y}_{\mathtt{s}_{[q_k]}\omega},\end{aligned}$$

we have

$$C^!\left(\left(\mathsf{Y}_{\mathtt{s}_{[p_1]}\omega} \underset{[p_2]}{\circ} \mathsf{Y}_{\mathtt{s}_{[p_2]}\omega}\right) \cdots \underset{[p_k]}{\circ} \mathsf{Y}_{\mathtt{s}_{[p_k]}\omega}\right) \simeq C^!\left(\left(\mathsf{Y}_{\mathtt{s}_{[q_1]}\omega} \underset{[q_2]}{\circ} \mathsf{Y}_{\mathtt{s}_{[q_2]}\omega}\right) \cdots \underset{[q_k]}{\circ} \mathsf{Y}_{\mathtt{s}_{[q_k]}\omega}\right).$$

Proof. By definition, the sequence $[p_1], \ldots, [p_k]$ (and likewise for $[q_1], \ldots, [q_k]$) has the following property: for $1 \leq i \leq j \leq k$, either $[p_i] \sqsubseteq [p_j]$ or $[p_i]$ and $[p_j]$ are $\sqsubseteq$-incomparable (recall that $\sqsubseteq$ is the prefix order on $\mathbb{A}_{n-1}$, see definition 2.41). Further, $\{[p_1], \ldots, [p_k]\} = \omega^\bullet = \{[q_1], \ldots, [q_k]\}$, i.e. the two sequences have the same elements. Consequently, the sequence $[q_1], \ldots, [q_k]$ can be obtained from $[p_1], \ldots, [p_k]$ by a series of transpositions of consecutive $\sqsubseteq$-incomparable addresses.

It is thus enough to check the following: for $\nu \in \mathbb{O}_n$, two different leaf addresses $[l], [l'] \in \nu^!$ (which are necessarily $\sqsubseteq$-incomparable), and $\psi, \psi' \in \mathbb{O}_{n-1}$ such that $\mathtt{t}\,\psi = \mathtt{e}_{[l]}\,\nu$ and $\mathtt{t}\,\psi' = \mathtt{e}_{[l']}\,\nu$, we have

$$C^!\left(\left(\nu \underset{[l]}{\circ} \mathsf{Y}_\psi\right) \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) = C^!\left(\left(\nu \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) \underset{[l]}{\circ} \mathsf{Y}_\psi\right).$$

Write

$$\begin{aligned} C^!(\nu) &= (E_\nu \triangleright \Gamma_\nu \vdash_n t_\nu : s_\nu \longmapsto X_\nu), \\ C^!(\mathsf{Y}_\psi) &= (E_\psi \triangleright \Gamma_\psi \vdash_n x_\psi : s_\psi \longmapsto X_\psi), \\ C^!(\mathsf{Y}'_\psi) &= (E_{\psi'} \triangleright \Gamma_{\psi'} \vdash_n x_{\psi'} : s_{\psi'} \longmapsto X_{\psi'}), \end{aligned}$$

with $t_\nu \in \mathbb{T}_n$ and $x_\psi, x_{\psi'} \in \mathbb{V}_n$. Let $a, a' \in (\mathtt{s}\,\nu)^\bullet$ be such that $\&_{\mathtt{s}\,\nu}\, a = [l]$ and $\&_{\mathtt{s}\,\nu}\, a' = [l']$ (see corollary 3.54). The sequents above are respectively obtained by the following proof trees:

$$\dfrac{\dfrac{\cdots \vdash t_\nu : s_\nu \longmapsto X_\nu \qquad \cdots \vdash x_\psi : s_\psi \longmapsto X_\psi}{F \triangleright \Gamma_\nu \cup \Gamma_\psi \vdash t_\nu(a \leftarrow x_\psi) : s_\nu[s_\psi/a] \longmapsto X_\nu}\,\texttt{graft-}a \qquad \cdots \vdash x_{\psi'} : s_{\psi'} \longmapsto X_{\psi'}}{G \triangleright \Gamma_\nu \cup \Gamma_\psi \cup \Gamma_{\psi'} \vdash t_\nu(a \leftarrow x_\psi)(a' \leftarrow x_{\psi'}) : s_\nu[s_\psi/a][s_{\psi'}/a'] \longmapsto X_\nu}\,\texttt{graft-}a'$$

$$\dfrac{\dfrac{\cdots \vdash t_\nu : s_\nu \longmapsto X_\nu \qquad \cdots \vdash x_{\psi'} : s_{\psi'} \longmapsto X_{\psi'}}{F' \triangleright \Gamma_\nu \cup \Gamma_{\psi'} \vdash t_\nu(a' \leftarrow x_{\psi'}) : s_\nu[s_{\psi'}/a'] \longmapsto X_\nu}\,\texttt{graft-}a' \qquad \cdots \vdash x_\psi : s_\psi \longmapsto X_\psi}{G' \triangleright \Gamma_\nu \cup \Gamma_{\psi'} \cup \Gamma_\psi \vdash t_\nu(a' \leftarrow x_{\psi'})(a \leftarrow x_\psi) : s_\nu[s_{\psi'}/a'][s_\psi/a] \longmapsto X_\nu}\,\texttt{graft-}a$$

It remains to prove that both those conclusions are α-equivalent.

(1) By assumption on the `graft` rule, $a \notin s^\bullet_{\psi'}$ and $a' \notin s^\bullet_\psi$, and clearly,

$$t_\nu(a \leftarrow x_\psi)(a' \leftarrow x_{\psi'}) = t_\nu(a' \leftarrow x_{\psi'})(a \leftarrow x_\psi).$$

(2) Again, since $a \notin s^\bullet_{\psi'}$ and $a' \notin s^\bullet_\psi$, we have $s_\nu[s_\psi/a][s_{\psi'}/a'] = s_\nu[s_{\psi'}/a'][s_\psi/a]$.

(3) Lastly, the equational theories G and G' are the union of E_ν, E_ψ, and $E_{\psi'}$, and the potential additional equalities incurred by the independent substitutions s_ψ/a and $s_{\psi'}/a'$. Hence $G = G'$. □

Corollary 3.63. *For any opetope $\omega \in \mathbb{O}$, the sequent $C^!(\omega)$ is uniquely defined up to α-equivalence.*

Proof. Clearly, proof trees (3.57), (3.58), and (3.59) are correct. In proposition 3.61, we have shown that the same holds for proof tree (3.60). Finally, in proposition 3.62, we have shown that for a non-degenerate opetope $\omega \in \mathbb{O}_n$, the sequent $C^!(\omega)$ does not depend on the decomposition of ω. □

Theorem 3.64. *(1) The polynomial coding $[\![-]\!]_n$ is a bijection between the set of derivable sequents typing an n-variable (up to α-equivalence), and $\mathbb{O}_n$. Its inverse is $C^!(-)$ restricted to $\mathbb{O}_n$.*

(2) If $n \geq 1$, the polynomial coding $[\![-]\!]_n$ is a bijection between the set of derivable sequents typing an $(n-1)$-terms (up to α-equivalence), and $\mathbb{O}_n$.

Proof. We prove point (1) first. The result is trivial if $n = 0, 1$, so we assume $n \geq 2$. We first show that for $\omega \in \mathbb{O}_n$ we have $\left[\!\left[C^!(\omega)\right]\!\right]_n = \omega$.

(1) By definition of $[\![-]\!]$, $\left[\!\left[C^!(\blacklozenge)\right]\!\right]_0 = \blacklozenge$.

(2) With the same notations as in (3.58), and by induction, we have

$$\left[\!\left[C^!\left(\mathsf{I}_\phi\right)\right]\!\right]_n = \mathsf{I}_{[\![C^!(\phi)]\!]_{n-2}} = \mathsf{I}_\phi.$$

(3) With the same notations as in (3.59), and by induction, we have,

$$\left[\!\left[C^!\left(\mathsf{Y}_\psi\right)\right]\!\right]_n = \mathsf{Y}_{[\![C^!(\psi)]\!]_{n-1}} = \mathsf{Y}_\psi.$$

(4) With the same notations as in (3.60), and by induction, we have

$$\begin{aligned}\left[\!\left[C^!\left(\nu \underset{[l]}{\circ} \mathsf{Y}_\psi\right)\right]\!\right]_n &= \left[\!\left[C^!(\nu)\right]\!\right]_n \underset{[\&_{\mathtt{s}\mathtt{s}\,u\,a}]}{\circ} \left[\!\left[C^!\left(\mathsf{Y}_\psi\right)\right]\!\right]_n \\ &= \left[\!\left[C^!(\nu)\right]\!\right]_n \underset{[l]}{\circ} \left[\!\left[C^!\left(\mathsf{Y}_\psi\right)\right]\!\right]_n \\ &= \nu \underset{[l]}{\circ} \mathsf{Y}_\psi.\end{aligned}$$

Conversely, we now show that for a derivable sequent $(E \triangleright \Gamma \vdash \alpha : T)$, we have an isomorphism $(E \triangleright \Gamma \vdash \alpha : T) \simeq C^!([\![E \triangleright \Gamma \vdash \alpha : T]\!]_n)$.

(1) We have that $C^!([\![x : \varnothing]\!]_0) = C^!(\blacklozenge) \simeq (\triangleright x : \varnothing \vdash x : \varnothing)$.

(2) With the same notations as in equation (3.45), we have

$$C^!([\![\cdots \vdash \delta : \underline{x} \mapsto x \mapsto X]\!]_n) = C^!\left(\mathsf{I}_{[\![x:X]\!]_n}\right)$$

and both sequents $C^!\left(\mathsf{I}_{[\![x:X]\!]_n}\right)$ and $(\cdots \vdash \delta : \underline{x} \mapsto x \mapsto X)$ are obtained by applying `degen` and `shift` to $(\cdots \vdash x : X)$. Thus $C^!([\![\cdots \vdash \delta : \underline{x} \mapsto x \mapsto X]\!]_n) \simeq (\cdots \vdash \delta : \underline{x} \mapsto x \mapsto X)$.

(3) Lastly, consider the sequent $\left(\cdots \vdash \alpha : x(\overrightarrow{y_i \leftarrow u_i}) \mapsto T\right)$ as in equation (3.46). Then

$$\begin{aligned}C^!\left(\left[\!\left[\cdots \vdash \alpha : x(\overrightarrow{y_i \leftarrow u_i}) \mapsto T\right]\!\right]_{n+1}\right) &= C^!\left(\mathsf{Y}_{[\![x]\!]_n} \bigcirc_{[\&_{\mathtt{s}\,x\,y_i}]} [\![u_i]\!]_{n+1}\right) \\ &\simeq \left(\cdots \vdash \alpha : x(\overrightarrow{y_i \leftarrow u_i}) \mapsto T'\right).\end{aligned}$$

Since T and T' are completely determined by $x(\overrightarrow{y_i \leftarrow u_i})$ (see theorem 3.31), we have that $T = T'$, whence

$$C^! \left(\left[\!\left[\cdots \vdash \alpha : x(\overrightarrow{y_i \leftarrow u_i}) \mapsto T \right]\!\right]_{n+1} \right) \simeq \left(\cdots \vdash \alpha : x(\overrightarrow{y_i \leftarrow u_i}) \mapsto T \right).$$

This concludes the proof of (1). Point (2) follows by noting that rule `shift` is a bijection up to α-equivalence between the set of derivable sequents typing an $(n-1)$-term and the set of derivable sequents typing an n-variable. □

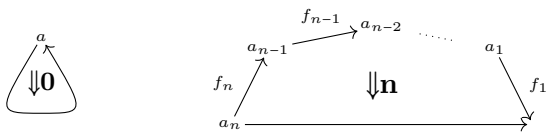
This concludes the inductive process started at definition 3.40.

3.3 Examples In this section, we showcase the derivation of some low dimensional opetopes. On a scale of a proof tree, specifying the context at every step is redundant. Hence we allow omitting it, only having the equational theory on the left of $\vdash$.

Example 3.65 (The arrow)**.** The unique 1-opetope, the *arrow*, is given by the following simple derivation:

$$\dfrac{\dfrac{}{\vdash_0 a : \varnothing}\,\texttt{point}}{\vdash_1 f : a \mapsto \varnothing}\,\texttt{shift}$$

Example 3.66 (Opetopic integers)**.** The opetopic integer $\mathbf{n}$ (example 2.36) is represented on the left in the case $n = 0$, and on the right if $n \geq 1$:



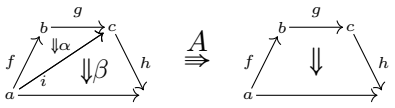
The derivation of $\mathbf{0}$ is

$$\dfrac{\dfrac{\dfrac{}{\vdash_0 a : \varnothing}\,\texttt{point}}{\vdash_1 \underline{a} : a \mapsto \varnothing}\,\texttt{degen}}{\vdash_1 \mathbf{0} : \underline{a} \mapsto a \mapsto \varnothing}\,\texttt{shift}$$

Alternatively, we could have used the `degen-shift` rule (remark 3.29). For $n \geq 1$, the opetope $\mathbf{n}$ is derived as follows, where `g` is a shorthand for `graft`:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots \quad \vdots}{\vdash f_1 : a_1 \mapsto \varnothing \qquad \vdash f_2 : a_2 \mapsto \varnothing}\,\texttt{g-}a_1 \qquad \vdots}{\vdash f_1(a_1 \leftarrow f_2) : a_2 \mapsto \varnothing \qquad \vdash f_3 : a_3 \mapsto \varnothing}\,\texttt{g-}a_2 \qquad \ddots}{\vdots \qquad\qquad\qquad \vdots}}{\vdash f_1(a_1 \leftarrow f_2(\cdots a_{n-2} \leftarrow f_{n-1})) : a_{n-1} \mapsto \varnothing \qquad \vdash f_n : a_n \mapsto \varnothing}\,\texttt{g-}a_{n-1}}{\vdash f_1(a_1 \leftarrow f_2(\cdots a_{n-1} \leftarrow f_n)) : a_n \mapsto \varnothing}\,\texttt{shift}}{\vdash \mathbf{n} : f_1(a_1 \leftarrow f_2(\cdots a_{n-1} \leftarrow f_n)) \mapsto a_n \mapsto \varnothing}$$

Example 3.67. The 3-opetope



is derived as follows:

$$\dfrac{\dfrac{\dfrac{}{\vdash_0 c : \varnothing}\,\texttt{point}}{\vdash_1 h : c \mapsto \varnothing}\,\texttt{shift} \quad \dfrac{\dfrac{}{\vdash_0 a : \varnothing}\,\texttt{point}}{\vdash_1 i : a \mapsto \varnothing}\,\texttt{shift}}{\vdash_1 h(c \leftarrow i) : c[a/c] \mapsto \varnothing}\,\texttt{graft-}c$$

and $c[a/c] = a$. Then,

$$\dfrac{\begin{array}{c}\vdots\\ \vdash_1 h(c \leftarrow i) : a \mapsto \varnothing\end{array}}{\vdash_2 \beta : h(c \leftarrow i) \mapsto a \mapsto \varnothing}\,\texttt{shift}$$

On the other hand we have

$$\dfrac{\dfrac{\dfrac{}{\vdash_0 b : \varnothing}\,\texttt{point}}{\vdash_1 g : b \mapsto \varnothing}\,\texttt{shift} \quad \dfrac{\dfrac{}{\vdash_0 a : \varnothing}\,\texttt{point}}{\vdash_1 f : a \mapsto \varnothing}\,\texttt{shift}}{\vdash_1 g(b \leftarrow f) : b[a/b] \mapsto \varnothing}\,\texttt{graft-}b$$

and $b[a/b] = a$. Then,

$$\dfrac{\begin{array}{c}\vdots\\ \vdash_2 \beta : h(c \leftarrow i) \mapsto a \mapsto \varnothing\end{array} \quad \dfrac{\begin{array}{c}\vdots\\ \vdash_1 g(b \leftarrow f) :\mapsto \varnothing\end{array}}{\vdash_2 \alpha : g(b \leftarrow f) \mapsto a \mapsto \varnothing}\,\texttt{shift}}{\vdash_2 \beta(i \leftarrow \alpha) : h(c \leftarrow i)[g(b \leftarrow f)/i] \mapsto a \mapsto \varnothing}\,\texttt{graft-}i$$

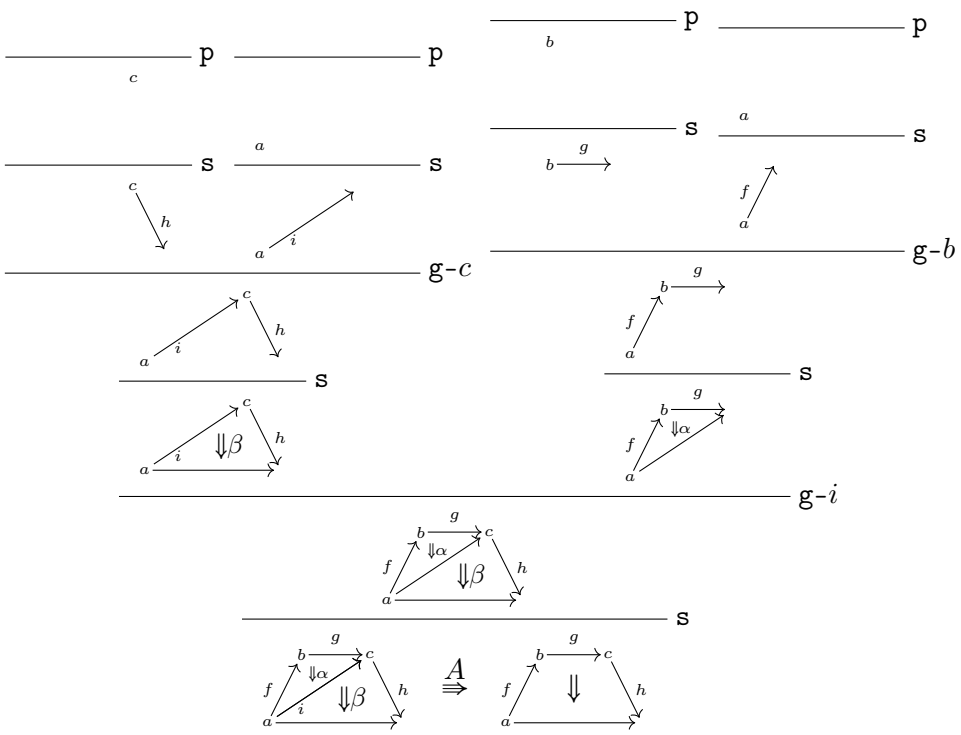
The last grafting is correct as $\mathtt{s}\, i = a = \mathtt{ss}\,\alpha$, and $h(c \leftarrow i)[g(b \leftarrow f)/i] = h(c \leftarrow g(b \leftarrow f))$. Finally

$$\dfrac{\begin{array}{c}\vdots\\ \vdash_2 \beta(i \leftarrow \alpha) : h(c \leftarrow g(b \leftarrow f)) \mapsto a \mapsto \varnothing\end{array}}{\vdash_3 A : \beta(i \leftarrow \alpha) \mapsto h(c \leftarrow g(b \leftarrow f)) \mapsto a \mapsto \varnothing}\,\texttt{shift}$$

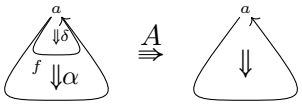
The complete proof tree is as follows, where `p`, `s`, and `g` are abbreviations for `point`, `shift`, and `graft`, respectively:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash c : \varnothing}\,\mathtt{p}}{\vdash h : c \mapsto \varnothing}\,\mathtt{s} \quad \dfrac{\dfrac{}{\vdash a : \varnothing}\,\mathtt{p}}{\vdash i : a \mapsto \varnothing}\,\mathtt{s}}{\vdash h(c \leftarrow i) : a \mapsto \varnothing}\,\mathtt{g}\text{-}c \quad \dfrac{\dfrac{\dfrac{}{\vdash b : \varnothing}\,\mathtt{p}}{\vdash g : b \mapsto \varnothing}\,\mathtt{s} \quad \dfrac{\dfrac{}{\vdash a : \varnothing}\,\mathtt{p}}{\vdash f : a \mapsto \varnothing}\,\mathtt{s}}{\vdash g(b \leftarrow f) : a \mapsto \varnothing}\,\mathtt{g}\text{-}b}{\vdash \beta : h(c \leftarrow i) \mapsto a \mapsto \varnothing}\,\mathtt{s} \quad \dfrac{}{\vdash \alpha : g(b \leftarrow f) \mapsto a \mapsto \varnothing}\,\mathtt{s}}{\vdash \beta(i \leftarrow \alpha) : h(c \leftarrow g(b \leftarrow f)) \mapsto a \mapsto \varnothing}\,\mathtt{g}\text{-}i}{\vdash A : \beta(i \leftarrow \alpha) \mapsto h(c \leftarrow g(b \leftarrow f)) \mapsto a \mapsto \varnothing}\,\mathtt{s}$$

This proof tree can be graphically represented as follows:



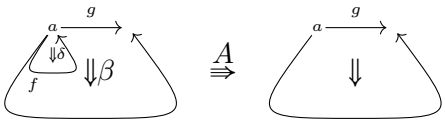
Example 3.68 (A degenerate case)**.** The 3-opetope



is derived as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash a : \varnothing}\,\texttt{point}}{\vdash f : a \mapsto \varnothing}\,\texttt{shift}}{\vdash \alpha : f \mapsto a \mapsto \varnothing}\,\texttt{shift} \qquad \dfrac{\dfrac{\dfrac{}{\vdash a : \varnothing}\,\texttt{point}}{\vdash \underline{a} : a \mapsto \varnothing}\,\texttt{degen}}{\vdash \delta : \underline{a} \mapsto a \mapsto \varnothing}\,\texttt{shift}}{\vdash \alpha(f \leftarrow \delta) : \underline{a} \mapsto a \mapsto \varnothing}\,\texttt{graft-}f}{\vdash A : \alpha(f \leftarrow \delta) \mapsto \underline{a} \mapsto a \mapsto \varnothing}\,\texttt{shift}$$

Example 3.69 (Another degenerate case)**.** The 3-opetope



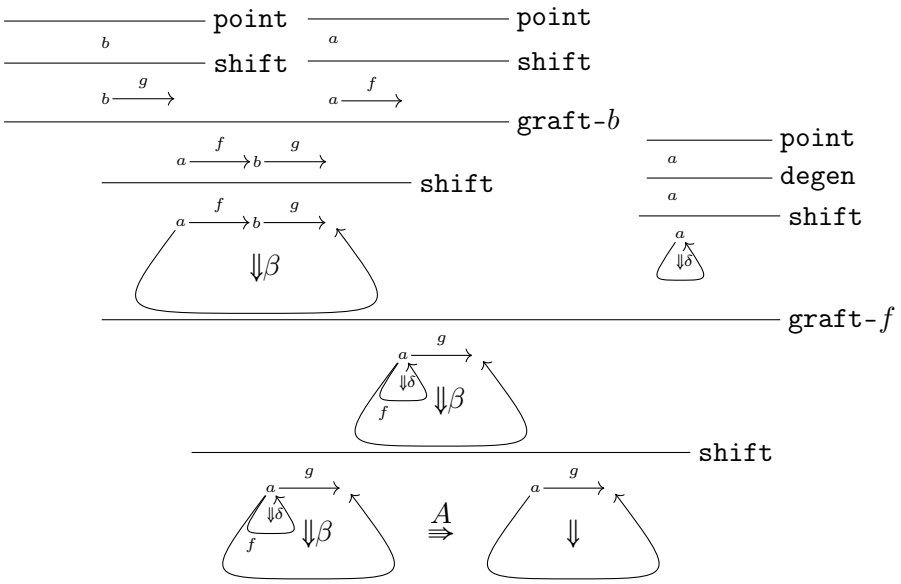
is derived as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash b : \varnothing}\,\texttt{point}}{\vdash g : b \mapsto \varnothing}\,\texttt{shift} \qquad \dfrac{\dfrac{}{\vdash a : \varnothing}\,\texttt{point}}{\vdash f : a \mapsto \varnothing}\,\texttt{shift}}{\vdash g(b \leftarrow f) : a \mapsto \varnothing}\,\texttt{graft-}b}{\vdash \beta : g(b \leftarrow f) \mapsto a \mapsto \varnothing}\,\texttt{shift} \qquad \dfrac{\dfrac{\dfrac{}{\vdash a : \varnothing}\,\texttt{point}}{\vdash \underline{a} : a \mapsto \varnothing}\,\texttt{degen}}{\vdash \delta : \underline{a} \mapsto a \mapsto \varnothing}\,\texttt{shift}}{a = b \vdash \beta(f \leftarrow \delta) : g(b \leftarrow f)[\underline{a}/f] \mapsto a \mapsto \varnothing}\,\texttt{graft-}f$$

and $g(b \leftarrow f)[\underline{a}/f] = g$, with the added equality $a = b$.

$$\dfrac{\begin{array}{c}\vdots\\ a = b \vdash \beta(f \leftarrow \delta) : g \longmapsto\bullet\, a \longmapsto\bullet\, \varnothing\end{array}}{a = b \vdash A : \beta(f \leftarrow \delta) \longmapsto\bullet\, g \longmapsto\bullet\, a \longmapsto\bullet\, \varnothing}\ \texttt{shift}$$

This proof tree can be graphically represented as follows:



3.4 Python implementation In this section, we briefly discuss the Python implementation [13] of the present work. System Opt! and all required syntactic constructs are implemented in `opetopy.NamedOpetope`. The rules are represented by functions `point`, `degen`, `shift`, `graft`, as well as `degen-shift` for the alternative rule presented in remark 3.29. Those rules are further encapsulated in rule instance classes `Point`, `Degen`, `shift`, `Graft`, and `DegenShift`, which represent rule instances in a proof tree, so constructing a derivation amounts to writing a Python term using those four classes. If that term evaluates without raising any exception, then the proof tree is considered correct.

Figure 3.1: Derivation of the arrow sequent using `opetopy.NamedOpetope`

```
from opetopy.NamedOpetope import *
# We first derive the point that will act as the source of the arrow by
 ↪  invoking the point rule on variable "a".
a = Point("a")
# We then apply the shift rule on a by providing a fresh variable, here "f".
f = Shift(a, "f")
# Since we use names, the following sequent, while corresponding to the same
 ↪  opetope, is different from f
g = Shift(Point("b"), "g")
# Note that the function opetopy.NamedOpetope.Arrow can be used to concisely
 ↪  get a proof tree of ■.
```

Figure 3.2: Derivation of some opetopic integers using `opetopy.NamedOpetope`, continuation of figure 3.1

```
opetopic_integer_0 = DegenShift(a, "n_0")
opetopic_integer_1 = Shift(f, "n_1")
opetopic_integer_2 = Shift(Graft(g, f, "b"), "n_2")
# Note that the function opetopy.NamedOpetope.OpetopicInteger can be used to
 ↪  get the proof tree of an arbitrary opetopic integer.
```

Figure 3.3: Derivation of example 3.67 using `opetopy.NamedOpetope`

```
from opetopy.NamedOpetope import *
f = Shift(Point("a"), "f")
g = Shift(Point("b"), "g")
h = Shift(Point("c"), "h")
i = Shift(Point("a"), "i")
alpha = Shift(Graft(g, f, "b"), "alpha")
beta = Shift(Graft(h, i, "c"), "beta")
A = Shift(Graft(beta, alpha, "i"), "A")
```

Figure 3.4: Derivation of example 3.69 using `opetopy.NamedOpetope`

```
from opetopy.NamedOpetope import *
f = Shift(Point("a"), "f")
g = Shift(Point("b"), "g")
alpha = DegenShift(Point("a"), "alpha")
beta = Shift(Graft(g, f, "b"), "beta")
D = Shift(Graft(beta, alpha, "f"), "D")
```

3.5 The system for opetopic sets We now present OptSet$^!$, a derivation system for opetopic sets that is based on Opt$^!$. We first present the required syntactic constructs and conventions in section 3.1.1, and present the inference rules in definition 3.71.

3.5.1 Syntax An interesting aspect of the named approach is that only the source faces are specified in the type of terms:

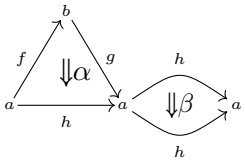
$$x : \mathtt{s}\, x \longmapsto \mathtt{ss}\, x \longmapsto \cdots$$

Nonetheless, as proven in proposition 3.53, all the information about targets remain. This comes from the intuition that any two opetopes with the same source are equal. In opetopic sets however, two cells with the same source faces need not be equal, nor have the same target. To adapt Opt$^!$ to opetopic sets, all faces, including targets, need to be explicitly specified. This will be part of rule `repr` of system OptSet$^!$, presented in definition 3.71. Lastly, recall that a typical sequent in system Opt$^!$ looks like this:

$$E \triangleright \Gamma \vdash t : T.$$

Here, t represents a pasting diagram that will ultimately serve as the source of a new variable, which will be introduced using rule `shift`. It does not provide any additional information about the variables of Γ and their adjacencies. Thus, when describing opetopic sets, we will drop the right hand side of the sequent, and deal with expressions of the form $(E \triangleright \Gamma)$, called *opetopic contexts modulo theory* (or OCMT for short).

Example 3.70. As a preliminary example, the OCMT describing the opetopic set:



is given by

$$\left(\begin{array}{ll} b = \mathtt{t}\, f, a = \mathtt{t}\, g = \mathtt{tt}\, \alpha = \mathtt{t}\, h = \mathtt{tt}\, \beta & a : \varnothing, b : \varnothing, \mathtt{t}\, f : \varnothing, \mathtt{t}\, g : \varnothing, \mathtt{tt}\, \alpha : \varnothing, \\ h = \mathtt{t}\, \beta = \mathtt{t}\, \alpha & \quad \mathtt{t}\, h : \varnothing, \mathtt{tt}\, \beta : \varnothing \\ & \triangleright \quad f : a \longmapsto \varnothing, g : a \longmapsto \varnothing, \mathtt{t}\, \alpha : a \longmapsto \varnothing, \\ & \quad h : a \longmapsto \varnothing, \mathtt{t}\, \beta : a \longmapsto \varnothing \\ & \alpha : g(b \leftarrow f) \longmapsto a \longmapsto \varnothing, \beta : h \longmapsto a \longmapsto \varnothing \end{array}\right)$$

On the right of $\triangleright$, we have all the variables and types, much like a context in Opt$^!$. The novelties are variables of the form $\mathtt{t}\, x$, which indicate targets. For example, $\mathtt{t}\, \alpha$, the target of α, has type $a \longmapsto \varnothing$, meaning that it is an arrow of source a. On the left hand side is an equational theory as in Opt$^!$, identifying variables. Note that $\mathtt{t}\, \alpha$ is identified with h, as shown by the diagram above. See example 3.104 for a full treatment of this case.

3.5.2 Inference rules Our derivation system for opetopic sets, presented in definition 3.71, has four rules:

(1) `repr` that takes an opetope in our previous system and turns it into the representable opetopic set of that opetope by adding all the target faces;
(2) `zero` that constructs the empty OCMT, corresponding to the empty opetopic set;
(3) `sum` that takes the disjoint union of two opetopic sets;
(4) `glue` that identifies cells of an opetopic set.

Every finite opetopic set is a quotient of a finite sum of representables. Therefore, those rules should be enough to derive all finite opetopic sets, which is formally demonstrated in theorem 3.101.

Definition 3.71 (The OptSet$^!$ system)**.** *Introduction of all targets.* This rule takes a sequent $(E \triangleright \Gamma \vdash_n x : X)$ typing a variable $x \in \mathbb{V}_n$, and completes it by adding all the targets cells, and turning it into an OCMT:

$$\frac{E \triangleright \Gamma \vdash_n x : X}{E' \triangleright \Gamma'}\ \texttt{repr}$$

where

$$\Gamma' := \Gamma \cup \left\{ \mathtt{t}^k\, a : \mathtt{s}^{k+1}\, a \longmapsto \mathtt{s}^{k+2}\, a \longmapsto \cdots \mid a \in \mathbb{V}_{\Gamma,l}, 1 \leq k \leq l \leq n \right\},$$

and

$$\begin{aligned} E' := E & \\ \cup\ & \{\mathtt{t}\, a = b \mid \text{for all } b \leftarrow a(\cdots) \text{ occurring in a type in } \Gamma\} && (3.72)\\ \cup\ & \left\{\mathtt{t}\mathtt{t}\, a = \mathtt{t}\mathtt{v}_{[]}\, \mathtt{s}\, a \mid a \in \mathbb{V}_{\Gamma',k}, \mathtt{s}\, a \text{ non degen.}, 2 \leq k \leq n\right\} && (3.73)\\ \cup\ & \left\{\mathtt{t}^2\, a = b \mid \text{if } a : \underline{b} \longmapsto b \longmapsto \cdots, a \in \mathbb{V}_{\Gamma',k} 2 \leq k\right\}. && (3.74) \end{aligned}$$

Here, $\mathtt{t}^k\, a = \mathtt{t} \cdots \mathtt{t}\, a$ can be thought of as a "tagging" on the variable $a \in \mathbb{V}_l$, but for simplicity, we consider it as a variable of its own: $\mathtt{t}^k\, a \in \mathbb{V}_{l-k}$. We also assume that variables of this form can only arise from applications of this rule. By convention, $\mathtt{t}^0\, a := a$, and if $a = b$, then $\mathtt{t}\, a = \mathtt{t}\, b$, for all $a, b \in \Gamma'$. In line (3.73), the source of a is assumed non-degenerate, thus $\mathtt{s}\, a$ is a term of the form $x(\overrightarrow{y_i \leftarrow u_i})$, and $\mathtt{v}_{[]}\, \mathtt{s}\, a = x$ (see definition 3.38).

Zero. This rules introduces the empty OCMT:

$$\frac{}{\triangleright}\ \texttt{zero}$$

Binary sums. This rule takes two disjoint opetopic sets (i.e. whose cells have different names), and produces their sum. If $\Gamma \cap \Upsilon = \varnothing$ (which implies that $E \cap F = \varnothing$), then

$$\frac{E \triangleright \Gamma \qquad F \triangleright \Upsilon}{E, F \triangleright \Gamma, \Upsilon}\ \texttt{sum}$$

Quotients. This rule identifies two parallel cells in an opetopic set by extending the underlying equational theory. If $a, b \in \mathbb{V}_\Gamma$ are such that $\mathtt{s}\, a =_E \mathtt{s}\, b$ and $\mathtt{t}\, a =_E \mathtt{t}\, b$, then

$$\frac{E \triangleright \Gamma}{E, a = b \triangleright \Gamma}\ \texttt{glue}$$

We also write `glue-`$(a{=}b)$ to make explicit that we added $\{a = b\}$ to the theory.

Remark 3.75. In rule `repr`, the additional equalities of (3.72) enforce **(Inner)**, those of (3.73) enforce **(Glob1)**, and those of (3.74) enforce **(Degen)**. Condition **(Glob2)** is implemented in (3.72), by the definition of the type of the target variables $\mathtt{t}\, a$: the bookkeeping of the readdressing map is completely transparent, as for an n-variable x, the correspondence between the $(n-1)$-variables of $\mathtt{s}\, x$ and $(n-1)$-variables of $\mathtt{st}\, x$ is already established by their name! See also remark 4.18.

Remark 3.76. Akin to Opt$^!$, in OptSet$^!$, an OCMT that is equivalent to a derivable one is itself derivable.

Remark 3.77. The `sum` and `zero` rules may be replaced by the following `usum` rule (unbiased sum) without changing the set of derivable OCMTs. For $k \geq 0$, and for $(E_1 \triangleright \Gamma_1), \ldots, (E_k \triangleright \Gamma_k)$ OCMTs such that $\Gamma_i \cap \Gamma_j = \varnothing$ for all $i \neq j$, then

$$\frac{E_1 \triangleright \Gamma_1 \quad \cdots \quad E_k \triangleright \Gamma_k}{E_1, \ldots, E_k \triangleright \Gamma_1, \ldots, \Gamma_k}\ \texttt{usum}$$

3.6 Equivalence with opetopic sets

3.6.1 Opetopic sets from OCTMs

Notation 3.78. Let $E \triangleright \Gamma$ be an OCMT. We write Γ/E for the set $\mathbb{V}_\Gamma$ quotiented by the equivalence relation generated by the equational theory E.

Definition 3.79 (OCMT of a variable)**.** For $(E \triangleright \Gamma \vdash_n x : X)$ a derivable sequent in system Opt$^!$, where $x \in \mathbb{V}_n$, let $(T_x \triangleright C_x)$, the *OCMT of* x (leaving the sequent around x implicit), be given by the following instance of `repr`:

$$\frac{E \triangleright \Gamma \vdash_n x : X}{T_x \triangleright C_x}\ \texttt{repr}$$

We now establish a series of results to prove that C_x/T_x carries a natural structure of representable opetopic set. We proceed in 4 steps:

(1) we start by noting that C_x/T_x is naturally a set over $\mathbb{O}$ via the polynomial coding map $[\![-]\!]$ (proposition 3.81);
(2) then, in proposition 3.83, we construct source and target maps in C_x/T_x, i.e. the structure maps of an opetopic set;
(3) we show in theorem 3.84 that the opetopic identities of definition 2.44 are satisfied, and that consequently, C_x/T_x has the structure of an opetopic set;
(4) finally, we show in proposition 3.91 that C_x/T_x is in fact a representable opetopic set, using a counting argument.

From there, we define a structure of opetopic set on an arbitrary OCMT by induction on its proof tree in definition 3.92.

Definition 3.80. Let $(E \triangleright \Gamma \vdash_n x : X)$ be a derivable sequent on Opt$^!$, where $x \in \mathbb{V}_n$, and $a \in \mathbb{V}_{C_x,k}$. If $a \in \mathbb{V}_{\Gamma,k}$, recall that by proposition 3.37, a is typed by the derivable sequent $(E|_a \triangleright \Gamma|_a \vdash_k a : A)$, where $(-)|_a$ denotes restriction of contexts and theories to a and to the variables occurring in the type A. Thus we have a well-defined opetope $[\![a]\!]_k = [\![E|_a \triangleright \Gamma|_a \vdash_k a : A]\!]_k \in \mathbb{O}_k$. Otherwise, if $a = \mathsf{t}^l\, b$ for some $b \in \mathbb{V}_{\Gamma,k+l}$, then $\mathsf{s}\, a = \mathsf{s}^{l+1}\, b$, and define $[\![a]\!]_k := \left[\!\left[\mathsf{s}^{l+1}\, b\right]\!\right]_k$. We thus have a map $[\![-]\!] : \mathbb{V}_{C_x} \longrightarrow \mathbb{O}$.

Proposition 3.81. *The map* $[\![-]\!] : \mathbb{V}_{C_x} \longrightarrow \mathbb{O}$ *factors through* C_x/T_x.

Proof. By construction, the theory T_x identifies variables $a, b \in \mathbb{V}_{C_x,k}$ only if $\mathsf{s}\, a = \mathsf{s}\, b$, thus $[\![a]\!]_k = [\![\mathsf{s}\, a]\!]_k = [\![\mathsf{s}\, b]\!]_k = [\![b]\!]_k$. □

Definition 3.82. For $\psi \in \mathbb{O}_k$, write

$$(C_x/T_x)_\psi \;=\; \left\{a \in \mathbb{V}_{C_x,k} \mid [\![a]\!]_k = \psi\right\}.$$

We now construct source and target maps between those subsets.

Sources. If $[p] \in [\![a]\!]^\bullet_k$, then by corollary 3.54, there is a unique $b \in \mathbb{V}_{C_x|_a,k-1}$ such that $\&_{\mathtt{s}\,a}\, b = [p]$. Let $\mathtt{v}_{[p]}\, a := b$.

Target. For $a \in \mathbb{V}_{C_x,k}$, $k > 0$, we set $\mathtt{t}(a) := \mathtt{t}\, a$, the latter being a variable introduced by the `repr` rule.

Proposition 3.83. *Let $a \in \mathbb{V}_{C_x,k}$.*

(1) For $[p] \in [\![a]\!]^\bullet_k$ we have $[\![\mathtt{v}_{[p]}\, \mathtt{s}\, a]\!]_{k-1} = \mathtt{s}_{[p]}\, [\![a]\!]_k$.

(2) We have $[\![\mathtt{t}\, a]\!]_{k-1} = \mathtt{t}\, [\![a]\!]_k$.

Proof. (1) It a is not a target i.e. $a \neq \mathtt{t}\, b$ for any $b \in \mathbb{V}_{C_x,k}$, then this is already proven by proposition 3.48. If $a = \mathtt{t}^l\, b$ for some $b \in \mathbb{V}_{C_x,k+l}$ that is not a target, and $l \in \mathbb{N}$, then

$$\begin{aligned}
[\![\mathtt{v}_{[p]}\, \mathtt{s}\, a]\!]_{k-1} &= [\![\mathtt{v}_{[p]}\, \mathtt{s}\mathtt{t}^l\, b]\!]_{k-1} \\
&= [\![\mathtt{v}_{[p]}\, \mathtt{s}^{l+1}\, b]\!]_{k-1} && \text{see definition of } \texttt{repr} \\
&= \mathtt{s}_{[p]}\, [\![\mathtt{s}\mathtt{s}^l\, b]\!]_k && \text{by proposition 3.48} \\
&= \mathtt{s}_{[p]}\, [\![\mathtt{s}\, a]\!]_k \\
&= \mathtt{s}_{[p]}\, [\![a]\!]_k && \text{see equation (3.43).}
\end{aligned}$$

(2) If a is not a target, then

$$\begin{aligned}
\mathtt{t}\, [\![a]\!]_k &= \mathtt{t}\, [\![\mathtt{s}\, a]\!]_k && \text{see equation (3.43)} \\
&= [\![\mathtt{s}\mathtt{s}\, a]\!]_{k-1} && \text{by proposition 3.53} \\
&= [\![\mathtt{s}\mathtt{t}\, a]\!]_{k-1} && \text{see definition of } \texttt{repr} \\
&= [\![\mathtt{t}\, a]\!]_{k-1} && \text{see equation (3.43).}
\end{aligned}$$

If $a = \mathtt{t}^l\, b$ for some $b \in \mathbb{V}_{C_x,k+l}$ that is not a target, and $l \in \mathbb{N}$, then

$$\begin{aligned}
\mathtt{t}\, [\![a]\!]_k &= \mathtt{t}\, [\![\mathtt{s}\, a]\!]_k && \text{see equation (3.43)} \\
&= \mathtt{t}\, [\![\mathtt{s}\mathtt{t}^l\, b]\!]_k \\
&= \mathtt{t}\, [\![\mathtt{s}^{l+1}\, b]\!]_k && \text{see definition of } \texttt{repr} \\
&= [\![\mathtt{s}^{l+2}\, b]\!]_{k-1} \\
&= \mathtt{t}\, [\![\mathtt{s}\mathtt{t}^{l+1}\, b]\!]_k && \text{see definition of } \texttt{repr} \\
&= [\![\mathtt{s}\mathtt{t}\, a]\!]_{k-1} \\
&= [\![\mathtt{t}\, a]\!]_{k-1} && \text{see equation (3.43).}
\end{aligned}$$

□

Theorem 3.84. *With all the structure of definition 3.82, C_x/T_x is an opetopic set.*

Proof. We check the opetopic identities of definition 2.44. Take $a \in \mathbb{V}_{C_x,k}$.

(Inner) Take $[p[q]] \in [\![a]\!]^\bullet_k$, and write $d = \mathtt{v}_{[p[q]]}\, \mathtt{s}\, a$. In $\mathtt{s}\, a$, the variable d occurs as

$$\mathtt{s}\, a \;=\; \cdots, b(\cdots, c \leftarrow d(\cdots), \cdots), \cdots$$

where $b = \mathtt{v}_{[p]}\, \mathtt{s}\, a$ and $c = \mathtt{v}_{[q]}\, \mathtt{s}\, b$. By equation (3.72), $\mathtt{v}_{[q]}\, \mathtt{v}_{[p]}\, a = \mathtt{v}_{[q]}\, b = c = \mathtt{t}\, d = \mathtt{t}\, \mathtt{v}_{[p[q]]}\, a$.

(Glob1) Assume that $\mathtt{s}\, a$ is not degenerate. Then, by equation (3.73), we have $\mathtt{t}\mathtt{t}\, a = \mathtt{t}\, \mathtt{v}_{[]}\, a$.

(Glob2) Assume that $\mathtt{s}\,a$ is not degenerate, take $[p[q]] \in [\![a]\!]^!_k$, and let $c := \mathtt{v}_{[q]}\,\mathtt{v}_{[p]}\,a$. Then

$$\begin{aligned}
\wp_{[\![a]\!]_k}[p[q]] &= \wp_{[\![\mathtt{s}a]\!]_k}[p[q]] && \text{see equation (3.43)}\\
&= \wp_{[\![\mathtt{s}a]\!]_k}\,\&_{\mathtt{s}\,a}\,c && \text{by definition}\\
&= \&_{\mathtt{ss}\,a}\,c && \text{by lemma 3.51}\\
&= \&_{\mathtt{st}\,a}\,c && \text{see definition of } \mathtt{repr}.
\end{aligned}$$

and thus $\mathtt{v}_{[q]}\,\mathtt{v}_{[p]}\,a = c = \mathtt{v}_{\&_{\mathtt{st}\,a}\,c}\,\mathtt{t}\,a = \mathtt{v}_{\wp_{[\![a]\!]_k}[p[q]]}\,\mathtt{t}\,a$.

(Degen) Assume that $\mathtt{s}\,a$ is degenerate, say $\mathtt{s}\,a = \underline{b}$. Then by equation (3.74), $\mathtt{v}_{[]}\,\mathtt{t}\,a = b = \mathtt{t}\,\mathtt{t}\,a = \mathtt{t}^0\,a = a$. □

Lemma 3.85. *The opetopic set C_x/T_x is a quotient of the representable $O[[\![x]\!]_n]$.*

Proof. The category of elements $\mathbb{O}/(C_x/T_x)$ of C_x/T_x is a direct category since $\mathbb{O}$ is too. It has a terminal object, namely the equivalence class of variable x itself. Moreover, that element is in the $[\![x]\!]_n$ component of C_x/T_x. By the Yoneda lemma, there is a map $f : O[[\![x]\!]_n] \longrightarrow C_x/T_x$ which induces a discrete fibration $\mathbb{O}/O[[\![x]\!]_n] = \mathbb{O}/[\![x]\!]_n \longrightarrow \mathbb{O}/(C_x/T_x)$ between the categories of elements. Since f has the terminal object x in its image, it is surjective on objects. □

Let $(E \triangleright \Gamma \vdash_n x : X)$ be a derivable sequent, with $x \in \mathbb{V}_n$. In lemma 3.85, we established that C_x/T_x is a quotient of the representable opetopic set $O[[\![x]\!]_n]$. We now aim to show that the two are actually isomorphic (proposition 3.91) by showing that they have the same number of cells.

Definition 3.86. for $\omega \in \mathbb{O}$, let

$$\#\omega := \sum_{\psi \in \mathbb{O}} \#O[\omega]_\psi = \sum_{\psi \in \mathbb{O}} \#\mathbb{O}(\psi, \omega),$$

which is a finite number since the slice category $\mathbb{O}/\omega$ is finite.

The strategy of the proof of proposition 3.91 is to show that the number of cells in C_x/T_x is precisely $\#[\![x]\!]_n$. We need some preliminary results first.

Proposition 3.87. *(1) We have $\#\blacklozenge = 1$, and $\#\blacksquare = 3$.*
(2) If ω is an endotope, say $\omega = \mathsf{Y}_\psi$, then $\#\omega = 2 + \#\psi$.
(3) If ω is a degenerate opetope, say $\omega = \mathsf{I}_\phi$, then $\#\omega = 2 + \#\phi$.
(4) If $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for some $\nu \in \mathbb{O}_n$, $[l] \in \nu^!$, and $\psi \in \mathbb{O}_{n-1}$, then $\#\omega = \#\nu + \#\psi - \#\,\mathsf{e}_{[l]}\,\nu$.

Proof. Point (1) is clear. We now prove point (2). Since source and target embeddings are generators of $\mathbb{O}$ (see definition 2.44), a non-identity morphism f with codomain ω necessarily factors through a source or the target embedding of ω; in this case, through $\mathsf{s}_{[]} : \psi \longrightarrow \omega$ or $\mathsf{t} : \psi \longrightarrow \omega$ (or both). However, by relations **(Glob1)** and **(Glob2)**, if f is not the target embedding of ω, then it necessarily factors through $\mathsf{s}_{[]} : \psi \longrightarrow \omega$. By inspection of the opetopic identities of definition 2.44, $\mathsf{s}_{[]} : \psi \longrightarrow \omega$ is a monomorphism. Indeed, it is impossible to paste the relation squares of definition 2.44 into one of the form

$$\begin{array}{ccc}
\phi & \xrightarrow{\mathsf{f}} & \mathsf{s}_{[]}\,\omega \\
{\scriptstyle \mathsf{g}}\downarrow & & \downarrow{\scriptstyle \mathsf{s}_{[]}} \\
\mathsf{s}_{[]}\,\omega & \xrightarrow[\mathsf{s}_{[]}]{} & \omega
\end{array}$$

with $\mathsf{f} \neq \mathsf{g}$. Consequently, as sets,

$$\mathbb{O}/\omega \;\cong\; \mathbb{O}/\psi + \{\mathsf{t} : \psi \longrightarrow \omega\} + \{\mathrm{id}_\omega : \omega \longrightarrow \omega\},$$

and the result follows. Points (3) and (4) follow the same type of argument. □

Corollary 3.88. *If $\omega \in \mathbb{O}_{\geq 2}$ is not degenerate, then*

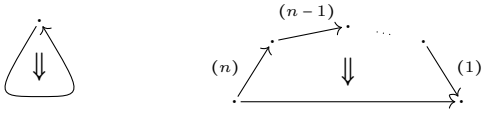
$$\#\omega = 2 + \left(\sum_{[p]\in\omega^\bullet} \#\,\mathsf{s}_{[p]}\,\omega\right) - \left(\sum_{[p[q]]\in\omega^\bullet} \#\,\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega\right)$$

Proof. If ω is an endotope, the result is already proved in proposition 3.87. Otherwise, decompose ω as $\nu \circ_{[l]} \mathsf{Y}_\psi$, and assume by induction that the result holds for ν. We have

$$\begin{aligned}
\#\omega &= \#\nu + \#\psi - \#\,\mathsf{e}_{[l]}\,\nu && \spadesuit\\
&= 2 + \left(\#\psi + \sum_{[p]\in\nu^\bullet} \#\,\mathsf{s}_{[p]}\,\nu\right) - \left(\#\,\mathsf{e}_{[l]}\,\nu + \sum_{[p[q]]\in\nu^\bullet} \#\,\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\nu\right) && \diamondsuit\\
&= 2 + \left(\#\,\mathsf{s}_{[l]}\,\omega + \sum_{\substack{[p]\in\omega^\bullet\\ [p]\neq[l]}} \#\,\mathsf{s}_{[p]}\,\omega\right) - \left(\#\,\mathsf{e}_{[l]}\,\omega + \sum_{\substack{[p[q]]\in\omega^\bullet\\ [p[q]]\neq[l]}} \#\,\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega\right)\\
&= 2 + \left(\sum_{[p]\in\omega^\bullet} \#\,\mathsf{s}_{[p]}\,\omega\right) - \left(\sum_{[p[q]]\in\omega^\bullet} \#\,\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\omega\right),
\end{aligned}$$

where ♠ is by proposition 3.87, and ♢ is by induction. □

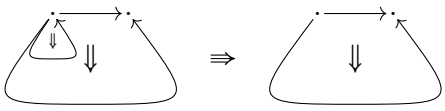
Example 3.89. Consider the opetopic integer $\mathbf{n} \in \mathbb{O}_2$ from example 2.36:



We show that $\#\mathbf{n} = 2n + 3$. If $n = 0$, then $\#\mathbf{0} = \#\mathsf{I}_\blacklozenge = 2 + \#\blacklozenge = 3$. This can be read on the graphical representation of $\mathbf{0}$, that has one point, one simple arrow, and one double arrow, for a total of 3 cells. If $n = 1$, then $\#\mathbf{1} = \#\mathsf{Y}_\blacksquare = 2 + \#\blacksquare = 5$. Otherwise,

$$\begin{aligned}
\#\mathbf{n} &= \#\left((\mathbf{n}-\mathbf{1}) \underset{[*^{n-1}]}{\circ} \mathsf{Y}_\blacksquare\right) && \text{by definition of } \mathbf{n}\\
&= \#(\mathbf{n}-\mathbf{1}) + \#\blacksquare - \#\,\mathsf{e}_{[*^{n-1}]}(\mathbf{n}-\mathbf{1}) && \text{by proposition 3.87}\\
&= (2n+1) + 3 - \#\blacklozenge && \text{by induction}\\
&= (2n+1) + 3 - 1 = 2n + 3.
\end{aligned}$$

Example 3.90. Consider the 3-opetope $\omega = \mathsf{Y}_{\mathbf{2}} \circ_{[[*]]} \mathsf{Y}_{\mathbf{0}}$ of example 3.69:



Then,

$$\begin{aligned}
\#\omega &= \#\left(\mathsf{Y_2} \underset{[[*]]}{\circ} \mathsf{Y_0}\right) \\
&= \#\mathsf{Y_2} + \#\mathbf{0} - \#\,\mathsf{e}_{[[*]]}\,\mathsf{Y_2} && \text{by proposition 3.87} \\
&= 2 + \#\mathbf{2} + \#\mathbf{0} - \#\blacksquare && \text{by proposition 3.87} \\
&= 9 && \text{since } \#\mathbf{n} = 2n+3.
\end{aligned}$$

Proposition 3.91. *We have* $C_x/T_x \cong O[[\![x]\!]_n]$.

Proof. (1) If $x = \blacklozenge$, then $\mathbb{V}_{C_x} = \blacklozenge$, while $T_x = \varnothing$. Thus $\#C_x/T_x = 1 = \#\blacklozenge$ by proposition 3.87. We know by lemma 3.85 that C_x/T_x is a quotient of $O[[\![x]\!]_n]$, and we just showed that the two have the same number of cells, namely $\#\blacklozenge = 1$. Consequently, $C_x/T_x \cong O[[\![x]\!]_0]$.

(2) Likewise, if $x = \blacksquare$, then $\mathbb{V}_{C_x} = \{\blacklozenge, \blacksquare, \mathtt{t}\blacksquare\}$, while $T_x = \varnothing$. Thus $\#C_x/T_x = 3 = \#\blacksquare$ by proposition 3.87, and by the argument above, $C_x/T_x \cong O[[\![x]\!]_1]$.

(3) Assume now that $x \in \mathbb{V}_n$ for $n \geq 2$. We proceed by case analysis on the form of $\mathtt{s}\,x$.

(a) If $\mathtt{s}\,x = y \in \mathbb{V}_{n-1}$, then $[\![x]\!]_n = \mathsf{Y}_{[\![y]\!]_{n-1}}$, and by proposition 3.87, $\#\,[\![x]\!]_n = 2 + \#\,[\![y]\!]_{n-1}$. Then $C_x = C_y + \left\{\mathtt{t}^k\,x \mid 0 \leq k \leq n\right\}$, and $\mathtt{t}\mathtt{t}\,x =_{T_x} \mathtt{t}\mathsf{v}_{[]}\,x = \mathtt{t}\,y$. Consequently, T_x is equivalent to the theory $T_y + \{\mathtt{t}\mathtt{t}\,x = \mathtt{t}\,y\}$, and thus

$$C_x/T_x \cong C_y/T_y + \{x, \mathtt{t}\,x\}\,.$$

By induction, $C_y/T_y \cong O[[\![y]\!]_{n-1}]$, and $\#C_x/T_x = 2 + \#C_y/T_y = 2 + \#\,[\![y]\!]_{n-1} = \#\,[\![x]\!]_n$, which, by the same argument as above, proves the isomorphism $C_x/T_x \cong O[[\![x]\!]_n]$.

(b) If $\mathtt{s}\,x = \underline{a}$ for some $a \in \mathbb{V}_{n-2}$, then $[\![x]\!]_n = \mathsf{I}_{[\![a]\!]_{n-2}}$, and by proposition 3.87, $\#\,[\![x]\!]_n = 2 + \#\,[\![a]\!]_{n-2}$. Then $C_x = C_a + \left\{\mathtt{t}^k\,x \mid 0 \leq k \leq n\right\}$, and $\mathtt{t}\mathtt{t}\,x =_{T_x} a$. Therefore T_x is equivalent to the theory $T_a + \{\mathtt{t}\mathtt{t}\,x = a\}$, and thus

$$C_x/T_x \cong C_a/T_a + \{x, \mathtt{t}\,x\}\,.$$

Consequently, $\#C_x/T_x = 2 + \#C_a/T_a = 2 + \#\,[\![a]\!]_{n-2} = \#\,[\![x]\!]_n$.

(c) Assume $\mathtt{s}\,x = t(a \leftarrow y)$, for some $t \in \mathbb{T}_{n-1}$, $a \in \mathbb{V}_{n-2}$, and $y \in \mathbb{V}_{n-1}$. Let $z : t \mapsto \cdots$ be a fresh n-variable. Clearly, $C_z - \{z, \mathtt{t}\,z\} \subseteq C_x$, thus

$$C_x = C_y \cup \left(C_z - \left\{\mathtt{t}^k\,z \mid 0 \leq k \leq n\right\}\right) + \left\{\mathtt{t}^k\,x \mid 0 \leq k \leq n\right\},$$

while T_x is equivalent to $T_y \cup T_z + \{\mathtt{t}\,y = a, \mathtt{t}\mathtt{t}\,x = \mathtt{t}\mathsf{v}_{[]}\,x\}$. Since $\mathsf{v}_{[]}\,\mathtt{s}\,x = \mathsf{v}_{[]}\,t = \mathsf{v}_{[]}\,\mathtt{s}\,z$, and $\mathtt{t}\mathsf{v}_{[]}\,\mathtt{s}\,z =_{T_z} \mathtt{t}\mathtt{t}\,z$ (see equation (3.73)), we have

$$C_x/T_x = C_y/T_y \cup C_z/T_z + \{x, \mathtt{t}\,x\} - \{z, \mathtt{t}\,z\}.$$

By hypothesis of the `graft` rule, $C_z/T_z \cap C_y/T_y = C_a/T_a$, and thus we have

$$\begin{aligned}
\#C_x/T_x &= \#C_y/T_y + \#C_z/T_z - \#C_a/T_a \\
&= \#\,[\![y]\!]_{n-1} + \#\,[\![z]\!]_n - \#\,[\![a]\!]_{n-2} \\
&= \#\left([\![t]\!]_n \underset{\&_t\, a}{\circ} \mathsf{Y}_{[\![y]\!]_{n-1}}\right) && \text{by proposition 3.87} \\
&= \#\,[\![x]\!]_n\,.
\end{aligned}$$

□

We now extend the structure of opetopic set defined in definition 3.82 to all OCMTs.

Definition 3.92. Let $(E \triangleright \Gamma)$ and $(F \triangleright \Upsilon)$ be two OCMTs, and assume by induction that Γ/E and Υ/F have a structure of opetopic set.

(1) If $(G \triangleright \Xi)$ is given by

$$\frac{E \triangleright \Gamma \qquad F \triangleright \Upsilon}{G \triangleright \Xi}\ \mathtt{sum}$$

then we have $\Xi = \Gamma + \Upsilon$, and $G = E + F$. There are natural set maps $i : \Gamma/E \longrightarrow \Xi/G$ and $j : \Upsilon/F \longrightarrow \Xi/G$, and clearly, there is a unique structure of opetopic set on Ξ/G making i and j into morphisms of opetopic sets. Further, Ξ/G is the coproduct of Γ/E and Υ/F, and i and j are the coprojections.

(2) Let $a, b \in \mathbb{V}_{\Gamma,k}$ be such that $\mathtt{s}\, a =_E \mathtt{s}\, b$ and $\mathtt{t}\, a =_E \mathtt{t}\, b$. Note that $O[\![a]\!]_k] = O[\![b]\!]_k]$. Then, by definition of rule `glue`, and for $(F \triangleright \Gamma)$ given by

$$\frac{E \triangleright \Gamma}{F \triangleright \Gamma}\ \mathtt{glue\text{-}}(a{=}b)$$

we have $F = E + \{a = b\}$. There is an obvious surjective set-map $p : \Gamma/E \longrightarrow \Gamma/F$, that identifies a and b, and preserves the other elements. From here, it is easy to see that there is a unique structure of opetopic set on Γ/F such that p is a morphism. In more details, if $x \in \Gamma/F$ is not the class of a or b, then its shape, sources, and target are the same as its unique preimage in Γ/E. Otherwise, since $\mathtt{s}\, a =_E \mathtt{s}\, b$ and $\mathtt{t}\, a =_E \mathtt{t}\, b$, a and b have the same shape, sources, and target (i.e. they are parallel), and so we can assign those of x accordingly.

Furthermore, it is easy to see that the following diagram exhibits p as the coequalizer of the maps a and b:

$$O[\![a]\!]_k] \underset{b}{\overset{a}{\rightrightarrows}} \Gamma/E \xrightarrow{\ p\ } \Gamma/F. \tag{3.93}$$

Proposition 3.94. *For $(E \triangleright \Gamma)$ a derivable OCMT in* OptSet$^!$*, the structure of opetopic set on Γ/E does not depend on the proof tree of $(E \triangleright \Gamma)$.*

Proof. If $\Gamma = \varnothing$, i.e. if the OCMT is obtained using the `zero` rule, then the result trivially holds. Otherwise, it is easy to see that the opetopic set Γ/E is given by the following expression that does not depend on the proof tree of $(E \triangleright \Gamma)$:

$$\Gamma/E \cong \frac{\sum_{k\in\mathbb{N}, a\in\mathbb{V}_{\Gamma,k}} O[\![a]\!]_k]}{a \sim b,\ \text{for all } a, b \in \mathbb{V}_\Gamma \text{ s.t. } a =_E b.}$$

By definition 3.80, for $a \in \mathbb{V}_\Gamma$, the opetope $[\![a]\!]_k$ only depends on the sequent. □

3.6.2 The equivalence Recall that $\widehat{\mathbb{O}}_{\text{fin}}$ is the full subcategory of $\widehat{\mathbb{O}}$ spanned by finite opetopic sets. In this subsection, we provide the last results needed to establish the equivalence between the category of derivable OCMTs and $\widehat{\mathbb{O}}_{\text{fin}}$.

Notation 3.95. For $(E \triangleright \Gamma)$ an OCMT, and $a, b \in \mathbb{V}$, the substitution $\Gamma[a/b]$ is defined in the obvious manner, by applying it to all typings in Γ.

Definition 3.96 (Morphism of OCMTs)**.** Let $(E \triangleright \Gamma)$ and $(F \triangleright \Upsilon)$ be OCMTs. A morphism $f : (E \triangleright \Gamma) \longrightarrow (F \triangleright \Upsilon)$ is a (non necessarily bijective) map $f : \mathbb{V}_\Gamma \longrightarrow \mathbb{V}_\Upsilon$ compatible with E and

F, such that if $x : X$ is a typing in Γ, then $f(x) : f(X)$ is a typing in Υ, where $f(X)$ is the result of applying f to every variable in X. Further, we require that for $n \geq 1$ and $x \in \mathbb{V}_{\Gamma,n}$, we have $f(\mathtt{t}\, x) = \mathtt{t}\, f(x)$. Note that this condition implies that f preserves the dimension of variables. Also, if $f, g : (E \triangleright \Gamma) \longrightarrow (F \triangleright \Upsilon)$, and if for all $x \in \mathbb{V}_\Gamma$ we have $f(x) =_F g(x)$, then we consider f and g to be equivalent, and only consider maps up to equivalence.

Lemma 3.97. *Morphisms of OCMTs preserve the shape of variables, i.e. for $f : (E \triangleright \Gamma) \longrightarrow (F \triangleright \Upsilon)$ a morphism and $a \in \mathbb{V}_{\Gamma,k}$, we have $[\![a]\!]_k = [\![f(a)]\!]_k$.*

Proof. Since there is a unique 0-opetope and a unique 1-opetope, the result holds trivially if $k = 0, 1$. If $k \geq 2$, we proceed by induction on $\mathtt{s}\, a$.

(1) If $\mathtt{s}\, a = b \in \mathbb{V}_{k-1}$, then $[\![a]\!]_k = \mathsf{Y}_{[\![b]\!]_{k-1}} = \mathsf{Y}_{[\![f(b)]\!]_{k-1}} = [\![f(a)]\!]_k$.
(2) If $\mathtt{s}\, a = \underline{b}$ for some $b \in \mathbb{V}_{k-2}$, then $[\![a]\!]_k = \mathsf{I}_{[\![b]\!]_{k-2}} = \mathsf{I}_{[\![f(b)]\!]_{k-2}} = [\![f(a)]\!]_{k-2}$.
(3) If $\mathtt{s}\, a = b(\overrightarrow{c_i \leftarrow u_i})$, then by induction, $[\![u_i]\!]_k = [\![f(u_i)]\!]_k$, and

$$[\![a]\!]_k \;=\; \mathsf{Y}_{[\![b]\!]_{k-1}} \bigcirc_{[\&_{\mathtt{s}\, b}\, c_i]} [\![u_i]\!]_k \;=\; [\![f(a)]\!]_k\,.$$

□

Definition 3.98. Let $\mathcal{C}\mathrm{tx}^!$ for the category of derivable OCMTs and such morphisms. In a sense, it is the syntactic category of system OptSet$^!$.

Definition 3.99 (Named stratification functor)**.** The *named stratification functor* $S^! : \mathcal{C}\mathrm{tx}^! \longrightarrow \widehat{\mathbb{O}}_{\mathrm{fin}}$ is defined as follows:

$$\begin{gathered} S^! : \mathcal{C}\mathrm{tx}^! \longrightarrow \widehat{\mathbb{O}}_{\mathrm{fin}} \\ (E \triangleright \Gamma) \longmapsto \Gamma/E \\ \Big((E \triangleright \Gamma) \xrightarrow{f} (F \triangleright \Upsilon)\Big) \longmapsto \Big(\Gamma/E \xrightarrow{S^! f} \Upsilon/F\Big). \end{gathered}$$

Proposition 3.100. *Let $f : (E \triangleright \Gamma) \longrightarrow (F \triangleright \Upsilon)$ be a morphism of OCMTs. Then the map $S^! f$ of definition 3.99 is indeed a morphism of opetopic sets.*

Proof. By definition, the cells of Γ/E are exactly the variables of $\mathbb{V}_\Gamma$, and likewise for Υ/F. By lemma 3.97, $S^! f$ preserves the shapes of the cells. By definition, if x is an n-variable, and $x : X$ a typing in Γ, then $f(x) : f(X)$ is a typing in Υ. So, for $\omega := [\![x]\!]_n$ and $[p] \in \omega^\bullet$, the following naturality square commutes:

$$\begin{array}{ccc} (\Gamma/E)_\omega & \xrightarrow{S^! f_\omega} & (\Upsilon/F)_\omega \\ {\scriptstyle \mathtt{s}_{[p]}}\downarrow & & \downarrow{\scriptstyle \mathtt{s}_{[p]}} \\ (\Gamma/E)_{\mathtt{s}_{[p]}\,\omega} & \xrightarrow{S^! f_{\mathtt{s}_{[p]}\,\omega}} & (\Upsilon/F)_{\mathtt{s}_{[p]}\,\omega}. \end{array}$$

By definition again, if $n \geq 1$, then $f(\mathtt{t}\, x) = \mathtt{t}\, f(x)$, so the analogous naturality square for target embeddings also commutes. Finally, $S^! f$ is a natural transformation. □

Theorem 3.101. *The stratification functor $S^! : \mathcal{C}\mathrm{tx}^! \longrightarrow \widehat{\mathbb{O}}_{\mathrm{fin}}$ is an equivalence of categories.*

Proof. The full subcategory of $\widehat{\mathbb{O}}_{\mathrm{fin}}$ spanned by the essential image of $S^!$ contains all the representables opetopic sets (proposition 3.91), the initial object (since $S^!(\triangleright)$ is the opetopic set with no cell), and is closed under finite sums and quotients (definition 3.92). Thus it is finitely cocomplete, and equal to the whole category $\widehat{\mathbb{O}}_{\mathrm{fin}}$, so $S^!$ is essentially surjective. By definition, $S^!$ is also faithful, and it remains to show that is it full.

Let $f : \Gamma/E \longrightarrow \Upsilon/F$ be a morphism of opetopic sets. Then, in particular, it is a map between the set of cells of Γ/E and Υ/F. To prove that it is a morphism of OCMT, we show that $\Gamma[f(x)/x \mid x \in \mathbb{V}_\Gamma]$ (see notation 3.95) is a subcontext of Υ modulo F, i.e. that for every typing $x : X$ in Γ, for some $x \in \mathbb{V}_k$, the type of $f(x)$ in Υ is $f(X)$ modulo F. If $(E \triangleright \Gamma)$ is the empty OCMT, the result is trivial. Let $x : X$ be a typing in Γ, with $x \in \mathbb{V}_k$. Since f is a morphism of opetopic sets, we have $f(x) \in \mathbb{V}_{\Upsilon,k}$, and

$$\begin{aligned} [\![x]\!]_k &= x^\natural && \text{in the opetopic set } \Gamma/E \\ &= f(x)^\natural \\ &= [\![f(x)]\!]_k\,. \end{aligned}$$

We show that the type of $f(x)$ in Υ is $f(X)$ by induction on k.

(1) If $k = 0$, then $X = \varnothing$. Since $f(x) \in \mathbb{V}_{\Upsilon,0}$, its type is necessarily $\varnothing = f(X)$, thus $f(x) : f(X)$ is a typing in Υ.

(2) If $k = 1$, then $X = (a \longmapsto \varnothing)$, where $a = \mathsf{v}_{[]}\, x$ in Γ/E, and since f is a morphism of opetopic sets, $f(\mathsf{v}_{[]}\, x) =_F \mathsf{v}_{[]}\, f(x)$. Thus

$$f(X) = \big(f(\mathsf{v}_{[]}\, x) \longmapsto \varnothing\big) =_F \big(\mathsf{v}_{[]}\, f(x) \longmapsto \varnothing\big)\,,$$

the latter being the type of $f(x)$ in Υ.

(3) Assume now that $k \geq 2$. The type of x is $X = (\mathsf{s}\,x \longmapsto \mathsf{ss}\,x \longmapsto \cdots \longmapsto \varnothing)$, and by definition, the type of $\mathsf{t}\,x$ is $Y := (\mathsf{ss}\,x \longmapsto \cdots \longmapsto \varnothing)$ (see equation (3.72)). By induction, the type of $f(\mathsf{t}\,x)$ in Υ is $f(Y)$, and since $f(\mathsf{t}\,x) =_F \mathsf{t}\,f(x)$, and the type of the latter is $(\mathsf{ss}\,f(x) \longmapsto \cdots \longmapsto \varnothing)$, we have

$$(f(\mathsf{ss}\,x) \longmapsto \cdots \longmapsto \varnothing) = f(Y) =_F (\mathsf{ss}\,f(x) \longmapsto \cdots \longmapsto \varnothing)\,,$$

or in other words, $\mathsf{s}^i\, f(x) =_F f(\mathsf{s}^i\, x)$, for $2 \leq i \leq k$. It remains to show that the latter formula holds in the case $i = 1$. Towards a contradiction, assume $\mathsf{s}\,f(x) \neq_F f(\mathsf{s}\,x)$. Then there exists $[p] \in [\![x]\!]_k^\bullet = [\![f(x)]\!]_k^\bullet$ such that $\mathsf{v}_{[p]}\, f(x) \neq_F f(\mathsf{v}_{[p]}\, x)$, which contradicts the fact that f is a morphism of opetopic sets. Consequently, $\mathsf{s}\,f(x) =_F f(\mathsf{s}\,x)$, and $f(X)$ is the type of $f(x)$ in Υ modulo F.

Finally, the underlying map of $f : \Gamma/E \longrightarrow \Upsilon/F$ is a morphism of OCMT, and $S^!$ is full. □

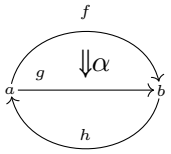
If $\mathcal{C}$ and $\mathcal{D}$ are small categories with finite limits, let $\mathcal{L}\mathrm{EX}(\mathcal{C}, \mathcal{D})$ be the category of left exact (i.e. finite limit preserving) functors from $\mathcal{C}$ to $\mathcal{D}$ and natural transformations. We now formally state the fact that system OPTSET$^?$ describe opetopic sets.

Theorem 3.102. *We have an equivalence* $\widehat{\mathbb{O}} \simeq \mathcal{L}\mathrm{EX}((\mathcal{C}\mathrm{tx}^!)^{\mathrm{op}}, \mathcal{S}\mathrm{et})$.

Proof. This follows directly from theorem 3.101 and from the Gabriel–Ulmer duality [6] [17, theorem 8]. □

3.7 Examples In this section, we give example derivations in system OPTSET$^!$. For clarity, we do not repeat the type of previously typed variables in proof trees.

Example 3.103. The opetopic set



is derived as follows. First, we derive the cells α, g, and h as opetopes (i.e. in OPTSET$^!$) to obtain the following sequents:

$$\begin{aligned} \triangleright a : \varnothing, f : a \longmapsto \varnothing, \alpha : f \longmapsto a \longmapsto \varnothing \vdash_2 \alpha : f \longmapsto a \longmapsto \varnothing \\ \triangleright c : \varnothing, g : c \longmapsto \varnothing \vdash_1 g : c \longmapsto \varnothing \\ \triangleright b : \varnothing, h : b \longmapsto \varnothing \vdash_1 h : b \longmapsto \varnothing \end{aligned}$$

and applying the `repr` rule yields respectively:

$$\begin{aligned} &\triangleright a : \varnothing, f : a \longmapsto \varnothing, \alpha : f \longmapsto a \longmapsto \varnothing, \mathtt{t}\, f : \varnothing, \mathtt{t}\, \alpha : a \longmapsto \varnothing, \mathtt{tt}\, \alpha : \varnothing \\ &\triangleright c : \varnothing, g : c \longmapsto \varnothing, \mathtt{t}\, g : \varnothing \\ &\triangleright b : \varnothing, h : b \longmapsto \varnothing, \mathtt{t}\, h : \varnothing. \end{aligned}$$

The proof tree then reads:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\triangleright a, f, \alpha \vdash_2 \alpha}}{\mathtt{tt}\,\alpha = \mathtt{t}\, f \triangleright a, f, \alpha, \mathtt{t}\, f, \mathtt{t}\,\alpha, \mathtt{tt}\,\alpha}\,\mathtt{repr} \quad \dfrac{\vdots}{\triangleright c, g \vdash_1 g}}{\triangleright c, g, \mathtt{t}\, g}\,\mathtt{repr}}{\mathtt{tt}\,\alpha = \mathtt{t}\, f \triangleright a, f, \alpha, \mathtt{t}\, f, \mathtt{t}\,\alpha, \mathtt{tt}\,\alpha, c, g, \mathtt{t}\, g}\,\mathtt{sum} \quad \dfrac{\vdots}{\triangleright b, h \vdash_1 h}}{\triangleright b, h, \mathtt{t}\, h}\,\mathtt{repr}}{\mathtt{tt}\,\alpha = \mathtt{t}\, f \triangleright a, f, \alpha, \mathtt{t}\, f, \mathtt{t}\,\alpha, \mathtt{tt}\,\alpha, c, g, \mathtt{t}\, g, \mathtt{t}\, h}\,\mathtt{sum}}{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}\,\mathtt{glue\text{-}}(a = c)$$

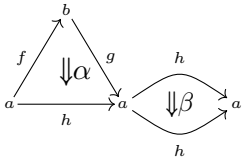
$$\dfrac{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}\,\mathtt{glue\text{-}}(b = \mathtt{t}\, f)$$

$$\dfrac{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f, b = \mathtt{t}\, g \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}\,\mathtt{glue\text{-}}(b = \mathtt{t}\, g)$$

$$\dfrac{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f, b = \mathtt{t}\, g \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f, b = \mathtt{t}\, g, a = \mathtt{t}\, h \\ \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}\,\mathtt{glue\text{-}}(a = \mathtt{t}\, h)$$

$$\dfrac{\begin{array}{l}\mathtt{tt}\,\alpha = \mathtt{t}\, f, a = c, b = \mathtt{t}\, f, b = \mathtt{t}\, g, a = \mathtt{t}\, h \\ g = \mathtt{t}\,\alpha \qquad \triangleright \\ \alpha\end{array} \quad \begin{array}{l} a, b, c, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{tt}\,\alpha \\ f, g, h, \mathtt{t}\,\alpha \end{array}}{}\,\mathtt{glue\text{-}}(g = \mathtt{t}\,\alpha)$$

Example 3.104. The opetopic set



is derived as follows. First, we derive the cells α and β as opetopes to obtain the following sequents:

$$\left(\begin{array}{ll} & a:\varnothing, b:\varnothing \\ \triangleright & f: a \longmapsto \bullet\, \varnothing, g: a \longmapsto \bullet\, \varnothing \\ & \alpha: g(b \leftarrow f) \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing \end{array} \quad \vdash_2 \quad \alpha: g(b \leftarrow f) \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing, \right)$$

$$\left(\begin{array}{ll} & a':\varnothing \\ \triangleright & h: a' \longmapsto \bullet\, \varnothing \\ & \beta: h \longmapsto \bullet\, a' \longmapsto \bullet\, \varnothing \end{array} \quad \vdash_2 \quad \beta: h \longmapsto \bullet\, a' \longmapsto \bullet\, \varnothing \right).$$

Applying the `repr` rule yields respectively:

$$\left(b = \mathtt{t} f, \mathtt{t} g = \mathtt{tt}\alpha \quad \triangleright \quad \begin{array}{l} a:\varnothing, b:\varnothing, \mathtt{t} f:\varnothing, \mathtt{t} g:\varnothing, \mathtt{tt}\alpha:\varnothing \\ f: a \longmapsto \bullet\, \varnothing, g: a \longmapsto \bullet\, \varnothing, \mathtt{t}\alpha: a \longmapsto \bullet\, \varnothing \\ \alpha: g(b \leftarrow f) \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing \end{array}\right),$$

$$\left(\mathtt{t} h = \mathtt{tt}\beta \quad \triangleright \quad \begin{array}{l} a':\varnothing, \mathtt{t} h:\varnothing, \mathtt{tt}\beta:\varnothing \\ h: a' \longmapsto \bullet\, \varnothing, \mathtt{t}\beta: a' \longmapsto \bullet\, \varnothing \\ \beta: h \longmapsto \bullet\, a' \longmapsto \bullet\, \varnothing \end{array}\right)$$

The proof tree then reads:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{a:\varnothing, b:\varnothing \;\triangleright\; f: a \longmapsto \bullet\, \varnothing, g: a \longmapsto \bullet\, \varnothing \;\vdash_2\; \alpha: g(b \leftarrow f) \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing}\,\mathtt{repr}}{b = \mathtt{t} f, \mathtt{t} g = \mathtt{tt}\alpha \;\triangleright\; a, b, \mathtt{t} f:\varnothing, \mathtt{t} g:\varnothing, \mathtt{tt}\alpha:\varnothing \;\; f, g, \mathtt{t}\alpha: a \longmapsto \bullet\, \varnothing \;\; \alpha}\,\mathtt{glue}\text{-}(a = \mathtt{t} g) \qquad \dfrac{\dfrac{\dfrac{\vdots}{a':\varnothing \;\triangleright\; h: a' \longmapsto \bullet\, \varnothing \;\vdash_2\; \beta: h \longmapsto \bullet\, a' \longmapsto \bullet\, \varnothing}\,\mathtt{repr}}{\mathtt{t} h = \mathtt{tt}\beta \;\triangleright\; a', \mathtt{t} h:\varnothing, \mathtt{tt}\beta:\varnothing \;\; h, \mathtt{t}\beta: a' \longmapsto \bullet\, \varnothing \;\; \beta}\,\mathtt{glue}\text{-}(h = \mathtt{t}\beta)}{\mathtt{t} h = \mathtt{tt}\beta,\; h = \mathtt{t}\beta \;\triangleright\; a', \mathtt{t} h, \mathtt{tt}\beta \;\; h, \mathtt{t}\beta \;\; \beta}\,\mathtt{glue}\text{-}(a' = \mathtt{t} h)}{b = \mathtt{t} f, a = \mathtt{t} g = \mathtt{tt}\alpha \;\triangleright\; a, b, \mathtt{t} f, \mathtt{t} g, \mathtt{tt}\alpha \;\; f, g, \mathtt{t}\alpha \;\; \alpha \qquad a', \mathtt{t} h, \mathtt{tt}\beta \;\triangleright\; h, \mathtt{t}\beta \;\; \beta}\,\mathtt{sum}$$

$$\dfrac{\dfrac{\dfrac{b = \mathtt{t} f, a = \mathtt{t} g = \mathtt{tt}\alpha, a' = \mathtt{t} h = \mathtt{tt}\beta,\; h = \mathtt{t}\beta \;\triangleright\; a, b, \mathtt{t} f, \mathtt{t} g, \mathtt{tt}\alpha, a', \mathtt{t} h, \mathtt{tt}\beta \;\; f, g, \mathtt{t}\alpha, h, \mathtt{t}\beta \;\; \alpha, \beta}{b = \mathtt{t} f, a = \mathtt{t} g = \mathtt{tt}\alpha = a' = \mathtt{t} h = \mathtt{tt}\beta,\; h = \mathtt{t}\beta \;\triangleright\; a, b, \mathtt{t} f, \mathtt{t} g, \mathtt{tt}\alpha, a', \mathtt{t} h, \mathtt{tt}\beta \;\; f, g, \mathtt{t}\alpha, h, \mathtt{t}\beta \;\; \alpha, \beta}\,\mathtt{glue}\text{-}(a = a')}{b = \mathtt{t} f, a = \mathtt{t} g = \mathtt{tt}\alpha = a' = \mathtt{t} h = \mathtt{tt}\beta,\; h = \mathtt{t}\beta = \mathtt{t}\alpha \;\triangleright\; a, b, \mathtt{t} f, \mathtt{t} g, \mathtt{tt}\alpha, a', \mathtt{t} h, \mathtt{tt}\beta \;\; f, g, \mathtt{t}\alpha, h, \mathtt{t}\beta \;\; \alpha, \beta}\,\mathtt{glue}\text{-}(\mathtt{t}\alpha = h)$$

Write $(E \triangleright \Gamma)$ for the final OCMT of this proof tree. At the beginning of section 3.5, we gave a seemingly different OCMT for the same opetopic set:

$$\left(\begin{array}{l} b = \mathtt{t} f, a = \mathtt{t} g = \mathtt{tt}\alpha = \mathtt{t} h = \mathtt{tt}\beta \\ h = \mathtt{t}\beta = \mathtt{t}\alpha \end{array} \quad \triangleright \quad \begin{array}{c} a:\varnothing, b:\varnothing, \mathtt{t} f:\varnothing, \mathtt{t} g:\varnothing, \mathtt{tt}\alpha:\varnothing, \\ \mathtt{t} h:\varnothing, \mathtt{tt}\beta:\varnothing \\ f: a \longmapsto \bullet\, \varnothing, g: a \longmapsto \bullet\, \varnothing, \mathtt{t}\alpha: a \longmapsto \bullet\, \varnothing, \\ h: a \longmapsto \bullet\, \varnothing, \mathtt{t}\beta: a \longmapsto \bullet\, \varnothing \\ \alpha: g(b \leftarrow f) \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing, \beta: h \longmapsto \bullet\, a \longmapsto \bullet\, \varnothing \end{array}\right).$$

But this is α-equivalent to $(E \triangleright \Gamma)$, as instances of a' have been replaced by a, which are both equal according to the equational theory E.

3.8 Python implementation The system OptSet$^!$ is implemented in the Python module `opetopy.NamedOpetopicSet` of [13]. The rules are represented by functions `repres` (since `repr` a Python standard function), `sum`, `glue`, and `zero`, and are further encapsulated in rule instance classes `Repr`, `Sum`, `Glue`, and `Zero`. We do not discuss the implementation, but we give an example derivation in figure 3.5.

Figure 3.5: Derivation of example 3.103 using `opetopy.NamedOpetopicSet`

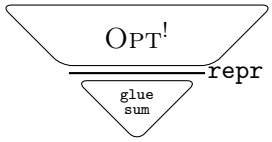
```
from opetopy.NamedOpetopicSet import *
# We first define all relevant variables using system Opt!.
f = Shift(Point("a"), "f")
g = Shift(Point("c"), "g")
h = Shift(Point("b"), "h")
alpha = Shift(f, "alpha")
# We then take the sum of all the representables we need. Note that the new
 ↪  target variables added by the repr rule have "t" prepended to their name
 ↪  e.g. the target variable of f is "tf", while that of α is "talpha".
example_unglued = Sum(Sum(Repr(alpha), Repr(g)), Repr(h))
example = Glue(
    Glue(
        Glue(
            Glue(
                Glue(
                    example_unglued, "a", "c"
                ), "b", "tf"
            ), "b", "tg"
        ), "a", "th"
    ), "g", "talpha"
)
```

3.9 The mixed system for opetopic sets The OptSet$^!$ system, presented in section 3.5, suffers from the following drawback: derivations of opetopic sets start with instances of rules `zero` or `repr`, the latter requiring a full opetope derivation in system Opt$^!$ (presented in section 3.1). This makes derivations somewhat unintuitive, since for an opetopic set $X \in \widehat{\mathbb{O}}$ written as

$$X = \frac{\sum_i O[\omega_i]}{\sim}$$

where $\sim$ represents some quotient, the opetopes ω_i have to be derived in Opt$^!$ first, then the `repr` rule has to be used on each one to produce the corresponding representables $O[\omega_i]$, and

only then can the sums and gluing be performed:



In this section, we present system $\textsc{OptSet}^!_{\textsc{m}}$ (the m standing for "mixed") for opetopic sets, which does not depend on $\textsc{Opt}^!$, and allows to perform introductions of new cells, sums, and gluings in any sound order. This is done by introducing new cells along with all their targets, effectively rendering $\textsc{OptSet}^!$'s `repr` rule superfluous, and removing the "barrier" between $\textsc{Opt}^!$ and $\textsc{OptSet}^!$ in the schema above.

3.9.1 Syntax The syntax of system $\textsc{OptSet}^!_{\textsc{m}}$ uses sequents from $\textsc{Opt}^!$ (see section 3.1) and OCMTs from $\textsc{OptSet}^!$. Specifically, we use two types of judgments.

(1) "$E \triangleright \Gamma$", stating that $E \triangleright \Gamma$ is a well formed OCMT.
(2) "$E \triangleright \Gamma \vdash t : T$", stating that in the OCMT $E \triangleright \Gamma$, the term t is well formed, and has type T. We may also write "$E \triangleright \Gamma \vdash_n t : T$" to emphasize that $t \in \mathbb{T}_n$.

3.9.2 Inference rules We present the inference rules of system $\textsc{OptSet}^!_{\textsc{m}}$ in definition 3.105. It uses rules `point`, and `graft` from system $\textsc{Opt}^!$, and rules `zero`, `sum`, and `glue` from system $\textsc{OptSet}^!$. Two new rules, `degen` and `pd`, will go from the first type of judgment to the second, by introducing degenerate terms and single variable terms respectively. In the other direction, rule `shift` is a variant of that of system $\textsc{Opt}^!$, and from $(E \triangleright \Gamma \vdash t : T)$, introduces a new cell having t as source, along with all the necessary targets. It can be viewed as a fusion of $\textsc{Opt}^!$'s `shift` rule and $\textsc{OptSet}^!$'s `repr` rule.

Definition 3.105 (The $\textsc{OptSet}^!_{\textsc{M}}$ system). *Introduction of points.* This rule introduces 0-cells, also called points. If $x \in \mathbb{V}_0$, then

$$\frac{}{\triangleright x : \varnothing}\ \texttt{point}$$

Introduction of degenerate pasting diagrams. This rule creates a new degenerate pasting diagram. If $x \in \mathbb{V}_{\Gamma,k}$, then

$$\frac{E \triangleright \Gamma, x : X}{E \triangleright \Gamma, x : X \vdash_{k+1} \underline{x} : x \longmapsto X}\ \texttt{degen}$$

Introduction of non-degenerate pasting diagrams. This rule creates a new non-degenerate pasting diagram consisting of a single cell. It can then be extended using the `graft` rule. If $x \in \mathbb{V}_{\Gamma,k}$, then

$$\frac{E \triangleright \Gamma, x : X}{E \triangleright \Gamma, x : X \vdash_k x : X}\ \texttt{pd}$$

Grafting. This rule extends a previously derived non-degenerate pasting diagram by grafting a cell. With the same conditions as rule `graft` of system $\textsc{Opt}^!$ (see section 3.1), if $x \in \mathbb{V}_n$, $t \in \mathbb{T}_n$ is not degenerate, $a \in (\mathtt{s}\, t)^\bullet$ is such that $\mathtt{s}\, a = \mathtt{ss}\, x$, then

$$\frac{E \triangleright \Gamma \vdash_n t : s_1 \longmapsto s_2 \longmapsto \cdots \qquad F \triangleright \Upsilon, x : X}{G \triangleright \Gamma \cup \Upsilon \vdash_n t(a \leftarrow x) : s_1[\mathtt{s}\, x/a] \longmapsto s_2 \longmapsto \cdots}\ \texttt{graft}$$

where G is the union of E, F, and potentially a set of additional equalities incurred by the substitution $s_1[\mathtt{s}\, x/a]$ (definition 3.21). We also write `graft-`a to make explicit the fact that we grafted onto a.

Shifting of pasting diagrams. This rule takes a previously derived pasting diagram (degenerate or not), and introduces a new cell having this pasting diagram as source. It also introduces the targets of all its iterated sources, and extends the ambient equational theory with the required identities, in the same fashion as rule `repr` of definition 3.23. If $x \in \mathbb{V}_{n+1}$ is such that $x \notin \mathbb{V}_\Gamma$, then

$$\frac{E \triangleright \Gamma \vdash_n t : T}{F \triangleright \Upsilon}\ \texttt{shift}$$

with

$$\Upsilon := \Gamma \cup \{x : t \longmapsto T\} \cup \left\{\mathtt{t}^i\, x : \mathtt{s}^{i+1}\, x \longmapsto \mathtt{s}^{i+2}\, x \longmapsto \cdots \mid 0 < i \leq n\right\},$$

where by convention, we let $\mathtt{t}^0\, x = x$, and F is defined as follows:

(1) if t is a degenerate term, say $t = \underline{a}$, then

$$F := E \cup \left\{\mathtt{t}^{i+2}\, x = \mathtt{t}^i\, a \mid 0 \leq i \leq n-1\right\} \tag{3.106}$$

(2) if t is not degenerate, say $t = y(\overrightarrow{z_i \leftarrow u_i})$, for some $y \in \mathbb{V}_n$, $\overrightarrow{z_i} \in \mathbb{V}_{n-1}$, and $\overrightarrow{u_i} \in \mathbb{T}_n$, then

$$\begin{aligned} F := {} & E \\ & \cup \left\{\mathtt{t}^2\, x = \mathtt{t}\, y \mid \text{if } n \geq 1\right\} \\ & \cup \left\{\mathtt{t}\, a = b \mid \text{for all } b \leftarrow a(\cdots) \text{ occurring in } t\right\}. \end{aligned}$$

Zero. This rules introduces the empty OCMT.

$$\frac{}{\triangleright}\ \texttt{zero}$$

Binary sums. This rule takes two disjoint OCMTs (i.e. whose cells have different names), and produces their sum. If $\Gamma \cap \Upsilon = \varnothing$, then

$$\frac{E \triangleright \Gamma \qquad F \triangleright \Upsilon}{E, F \triangleright \Gamma, \Upsilon}\ \texttt{sum}$$

Quotients. This rule identifies two parallel cells in an opetopic set by extending the underlying equational theory. If $a, b \in \mathbb{V}_\Gamma$ are such that $\mathtt{s}\, a =_E \mathtt{s}\, b$ and $\mathtt{t}\, a =_E \mathtt{t}\, b$, then

$$\frac{E \triangleright \Gamma}{E, a = b \triangleright \Gamma}\ \texttt{glue}$$

We also write $\texttt{glue-}(a{=}b)$ to make explicit that we added $a = b$ to the theory.

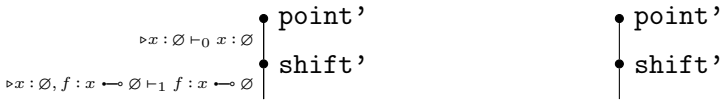
Remark 3.107. The `sum` and `zero` rules may be replaced by the following `usum` rule (unbiased sum) without changing the set of derivable OCMTs. For $k \geq 0$, and for $(E_1 \triangleright \Gamma_1), \ldots, (E_k \triangleright \Gamma_k)$ OCMTs such that $\Gamma_i \cap \Gamma_j = \varnothing$ for all $i \neq j$, then

$$\frac{E_1 \triangleright \Gamma_1 \qquad \cdots \qquad E_k \triangleright \Gamma_k}{E_1, \ldots, E_k \triangleright \Gamma_1, \ldots, \Gamma_k}\ \texttt{usum}$$

Remark 3.108. Akin to Opt$^!$ and OptSet$^!$, in OptSet$^!_{\mathrm{M}}$ a sequent or an OCMT that is equivalent to a derivable one is itself derivable.

3.10 Equivalence with opetopic sets The aim of this section is to prove theorem 3.113, stating that system OptSet$^!_{\mathrm{M}}$ precisely derives opetopic sets, in the sense of theorems 3.101 and 3.102. In other words, we prove that the set of derivable OCMTs of systems OptSet$^!_{\mathrm{M}}$ and OptSet$^!$ are the same. This is done by rewriting proof trees in OptSet$^!$ to proof trees in OptSet$^!_{\mathrm{M}}$ (see proposition 3.110) and conversely (see proposition 3.112).

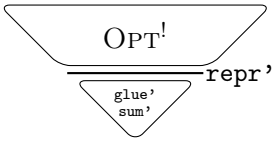
Convention 3.109. Throughout this section, the rules of systems Opt$^!$ and OptSet$^!$ will be decorated by a prime, e.g. `shift'`, in order to differentiate them from the rules of system OptSet$^!_{\mathrm{M}}$. Further, to make notations lighter and the demonstrations more graphical, we write proof trees as actual trees, whose nodes are decorated by rules, and edges by sequents or OCMTs. For instance, derivation of the arrow $\blacksquare$ (see example 3.65) in system Opt$^!$ is represented as on the left, or more concisely as on the right:



If no uncertainty arises, we leave the decoration of the edges implicit, as on the right.

Proposition 3.110. *Every OCMT derivable in system* OptSet$^!$ *is also derivable in system* OptSet$^!_{\mathrm{M}}$.

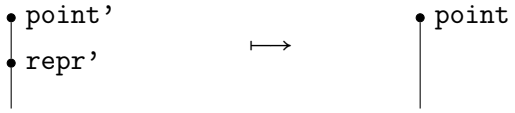
Proof. Recall that a proof tree in system OptSet$^!$ has the following structure:



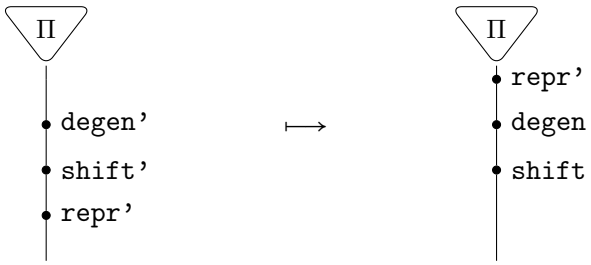
meaning that it begins with derivations in system Opt$^!$, followed by instances of the `repr'` rule, followed by a derivation in system OptSet$^!$. Remark that rule `glue'` is exactly `glue`, and likewise for `sum`, so that the bottom part of the proof tree is already a derivation in system OptSet$^!_{\mathrm{M}}$.

We now show that we can rewrite the top part to a proof in system OptSet$^!_{\mathrm{M}}$ by "moving up " the instances of rule `repr`, and replacing the other rule instances by those of OptSet$^!_{\mathrm{M}}$. This rewriting procedure is defined by the following cases.

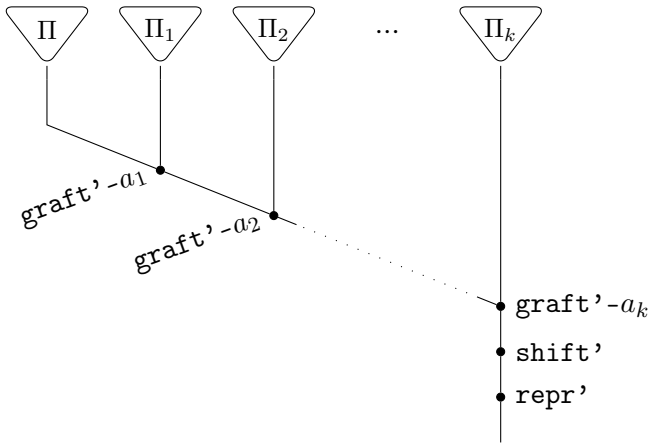
(1) If we have a proof tree as on the left, we rewrite it as on the right:

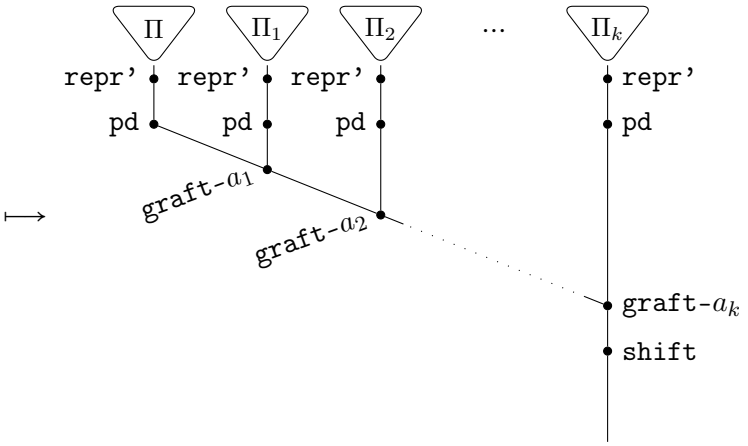


(2) If we have a derivation as on the left, where Π is a proof tree in system Opt$^!$ or OptSet$^!$, then we rewrite it as on the right:



(3) If $k \geq 0$ and if we have a derivation as on the top, where $\Pi, \Pi_1, \ldots, \Pi_k$ are proof trees in system Opt$^!$ or OptSet$^!$, then we rewrite it as below:

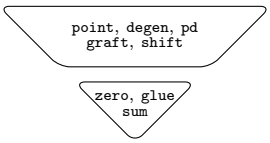




Here, the new instances of `pd` pick the adequate variables from each sequent, so that they can be used by the instances of `graft`. Once the grafting process is complete, rule `shift` adds all necessary targets, which was previously done by `repr'`.

It is routine verification to check that the conclusion OCMT on the left and the right of any of those cases are the same. This rewriting procedure is terminating, and a normal form of a proof tree in system OptSet$^!$ is a proof tree in system OptSet$^!_M$ that derives the same OCMT. □

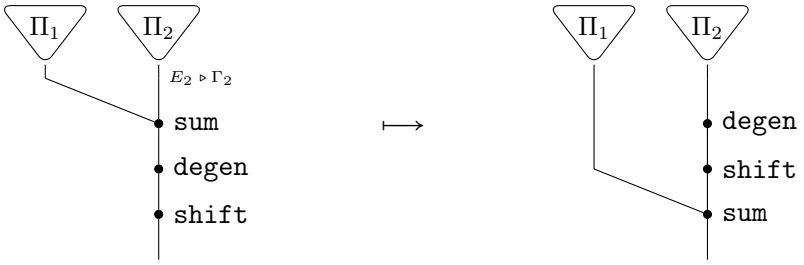
Lemma 3.111. *Let* $(E \triangleright \Gamma)$ *be a derivable OCMT in system* OptSet$^!_M$*. Then it admits a proof tree of the following form*



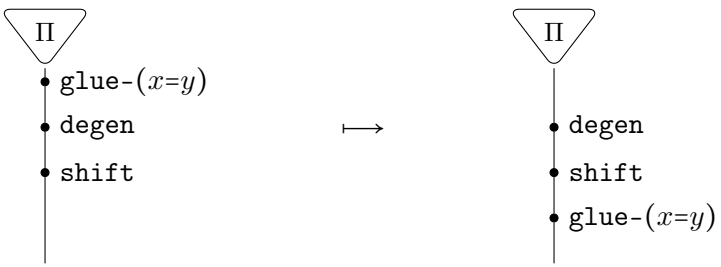
meaning a proof tree starting with a derivation in the fragment of system OptSet$^!_M$ *containing only rules* `point`*,* `degen`*,* `pd`*,* `graft`*, and* `shift`*, followed by a derivation in the complementary fragment.*

Proof. If we have a proof tree consisting only of an instance of rule `zero`, then the result trivially holds. Otherwise, we proceed by stating rewriting steps of proof trees in system OptSet$^!_M$, as in the proof of proposition 3.110.

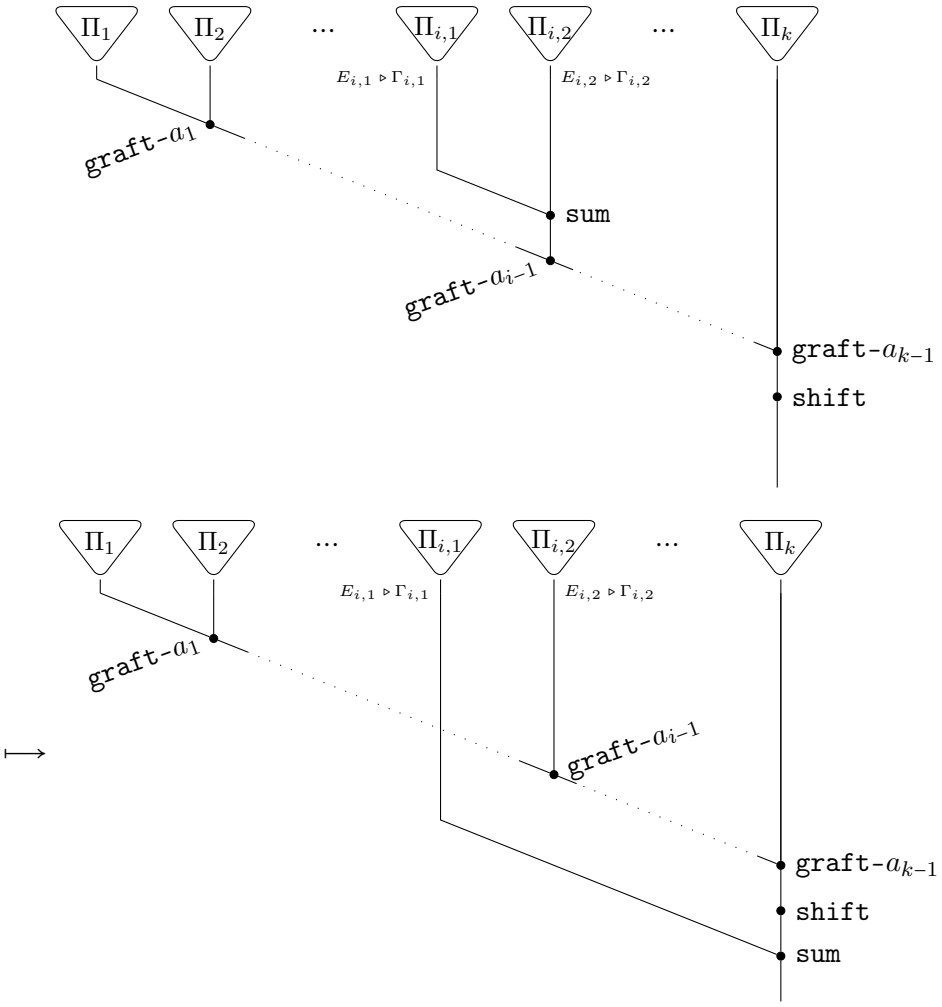
(1) If we have a proof tree as on the left, and assuming the instance of `degen` degenerates a variable in Γ_2, we rewrite it as on the right:



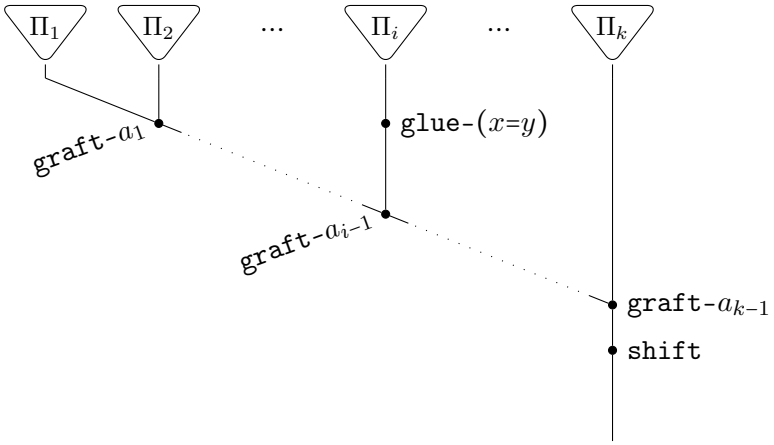
(2) A proof tree as on the left is rewritten as on the right:

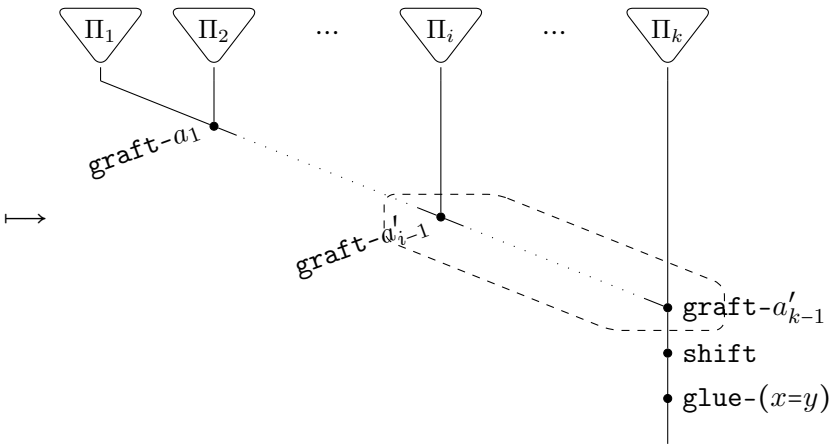


(3) Consider a proof tree as on the top. Then, by assumption on rule `sum`, either $a_{i-1} \in \mathbb{V}_{\Gamma_1}$ or $a_{i-1} \in \mathbb{V}_{\Gamma_2}$. Without loss of generality, assume the latter holds. Then we rewrite the proof tree as below:



(4) Consider a proof tree as on the top, and rewrite it as below, x.





where in the dashed zone, for $i-1 \leq j \leq k-1$,

$$a'_j = \begin{cases} x & \text{if } a_j = y \\ a_j & \text{otherwise.} \end{cases}$$

In other words, the uses of y in the dashed zone have been replaced by uses of x. □

Proposition 3.112. *Every OCMT derivable in system* $\textsc{OptSet}^!_{\mathrm{M}}$ *is also derivable in system* $\textsc{OptSet}^!$.

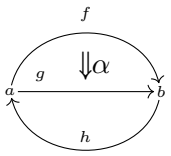
Proof. Consider a proof tree in system $\textsc{OptSet}^!_{\mathrm{M}}$. Then it can be rewritten so as to have the shape described in lemma 3.111. Applying the rewriting steps of proposition 3.110 in reverse order yields a proof tree in systems $\textsc{Opt}^!$ and $\textsc{OptSet}^!$ that derives the same OCMT. □

Theorem 3.113. *The system* $\textsc{OptSet}^!_{\mathrm{M}}$ *derives opetopic sets in the sense of theorems 3.101 and 3.102.*

Proof. By propositions 3.110 and 3.112, the OCMTs derived by system $\textsc{OptSet}^!_{\mathrm{M}}$ and $\textsc{OptSet}^!$ are the same. □

3.11 Examples In this section, we give example derivations in system $\textsc{OptSet}^!_{\mathrm{M}}$. For clarity, we do not repeat the type of previously typed variables in proof trees.

Example 3.114. The opetopic set



of example 3.103 can be derived as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\triangleright\ a : \varnothing}\,\texttt{point}}{\triangleright\ a \vdash_0 a}\,\texttt{pd}}{\triangleright\ \begin{array}{l} a, \mathtt{t}\, f : \varnothing \\ f : a \longmapsto \varnothing \end{array}}\,\texttt{shift}}{\triangleright\ \begin{array}{l} a, \mathtt{t}\, f \\ f \end{array} \vdash_0 a}\,\texttt{pd} \qquad \dfrac{\dfrac{\dfrac{}{\triangleright\ b : \varnothing}\,\texttt{point}}{\triangleright\ b \vdash_0 b}\,\texttt{pd}}{\triangleright\ \begin{array}{l} b, \mathtt{t}\, h : \varnothing \\ h : b \longmapsto \varnothing \end{array}}\,\texttt{shift}$$

$$\dfrac{\triangleright\ \begin{array}{l} a, \mathtt{t}\, f \\ f, g : a \longmapsto \varnothing \end{array} \qquad \triangleright\ \begin{array}{l} b, \mathtt{t}\, h : \varnothing \\ h : b \longmapsto \varnothing \end{array}}{\triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array}}\,\texttt{sum}$$

$$\dfrac{\triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array}}{b = \mathtt{t}\, f \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array}}\,(b{=}\mathtt{t}\, f)$$

$$\dfrac{b = \mathtt{t}\, f \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array}}{b = \mathtt{t}\, f = \mathtt{t}\, g \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array}}\,(b{=}\mathtt{t}\, g)$$

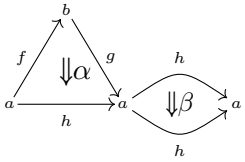
$$\dfrac{b = \mathtt{t}\, f = \mathtt{t}\, g \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h \\ f, g, h \end{array} \quad \vdash_1 f}{b = \mathtt{t}\, f = \mathtt{t}\, g = \mathtt{t}^2\, \alpha \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{t}^2\, \alpha : \varnothing \\ f, g, h, \mathtt{t}\, \alpha : a \longmapsto \varnothing \\ \alpha : f \longmapsto a \longmapsto \varnothing \end{array}}\,\texttt{pd}\ \texttt{shift}$$

$$\dfrac{b = \mathtt{t}\, f = \mathtt{t}\, g = \mathtt{t}^2\, \alpha \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{t}^2\, \alpha : \varnothing \\ f, g, h, \mathtt{t}\, \alpha : a \longmapsto \varnothing \\ \alpha : f \longmapsto a \longmapsto \varnothing \end{array}}{\begin{array}{l} b = \mathtt{t}\, f = \mathtt{t}\, g = \mathtt{t}^2\, \alpha \\ g = \mathtt{t}\, \alpha \end{array} \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{t}^2\, \alpha : \varnothing \\ f, g, h, \mathtt{t}\, \alpha : a \longmapsto \varnothing \\ \alpha : f \longmapsto a \longmapsto \varnothing \end{array}}\,(g{=}\mathtt{t}\, \alpha)$$

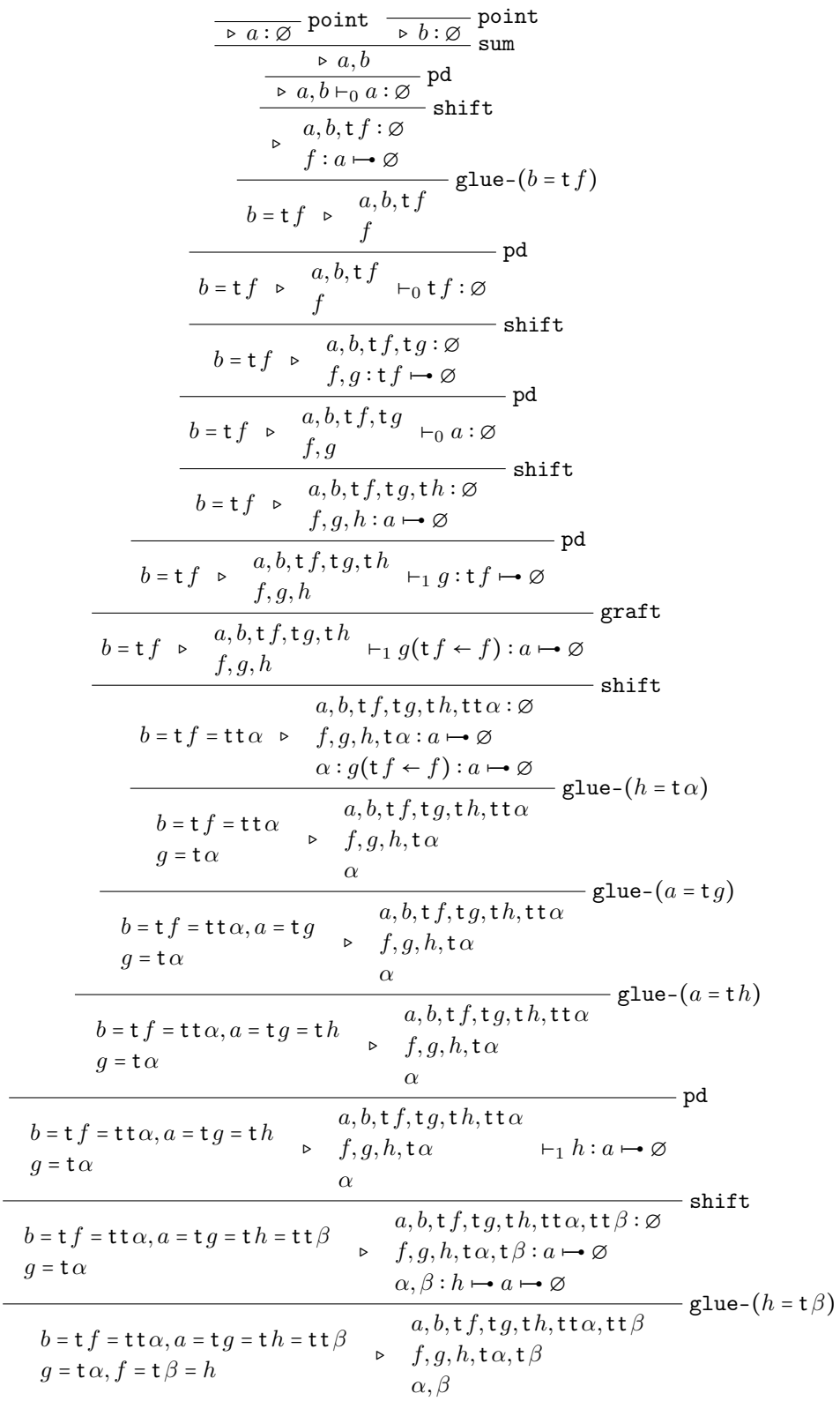
$$\dfrac{\begin{array}{l} b = \mathtt{t}\, f = \mathtt{t}\, g = \mathtt{t}^2\, \alpha \\ g = \mathtt{t}\, \alpha \end{array} \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{t}^2\, \alpha : \varnothing \\ f, g, h, \mathtt{t}\, \alpha : a \longmapsto \varnothing \\ \alpha : f \longmapsto a \longmapsto \varnothing \end{array}}{\begin{array}{l} b = \mathtt{t}\, f = \mathtt{t}\, g = \mathtt{t}^2\, \alpha, a = \mathtt{t}\, h \\ g = \mathtt{t}\, \alpha \end{array} \quad \triangleright\ \begin{array}{l} a, b, \mathtt{t}\, f, \mathtt{t}\, g, \mathtt{t}\, h, \mathtt{t}^2\, \alpha : \varnothing \\ f, g, h, \mathtt{t}\, \alpha : a \longmapsto \varnothing \\ \alpha : f \longmapsto a \longmapsto \varnothing \end{array}}\,(a{=}\mathtt{t}\, h)$$

where rules of the form $(x = y)$ are shorthands for $\texttt{glue-}(x = y)$.

Example 3.115. The opetopic set



of example 3.104 can be derived as follows.



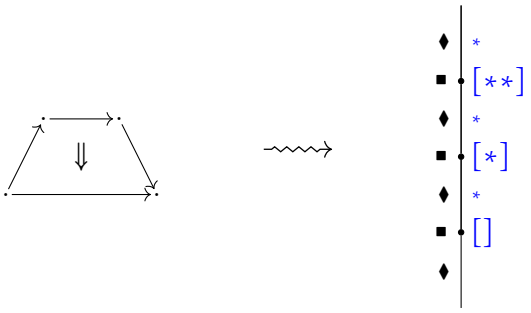
3.12 Python implementation The system $\textsc{OptSet}^{!}_{\mathrm{m}}$ is implemented in the Python module `opetopy.NamedOpetopicSetM` of [13]. Its usage is very similar to `opetopy.NamedOpetope` and `opetopy.NamedOpetopicSet`, presented in sections 3.4 and 3.8 respectively.

Figure 3.6: Derivation of example example 3.103 using `opetopy.NamedOpetopicSetM`

```
from NamedOpetopicSetM import Glue, Pd, Point, RuleInstance, Shift, Sum

# We first derive the f, g components
p1 = Shift(Pd(Point("a"), "a"), "f")
p1 = Shift(Pd(p1, "a"), "g")

# We then derive the h component
p2 = Shift(Pd(Point("b"), "b"), "h")

# We proceed to sum the two, glue some cells, and introduce alpha
example = Sum(p1, p2)  # type: RuleInstance
example = Glue(example, "b", "tf")
example = Glue(example, "b", "tg")
example = Shift(Pd(example, "f"), "alpha")
example = Glue(example, "b", "ttalpha")
example = Glue(example, "g", "talpha")
example = Glue(example, "a", "th")
```

4. Unnamed approach

4.1 The system for opetopes The unnamed approach for opetopes relies on the calculus of higher addresses presented in section 2.2.3 to identify cells, rather than on names as in the named approach. For example, recall the opetope $\mathbf{3} \in \mathbb{O}_2$ from example 2.36, drawn on the left, with its underlying $\mathfrak{Z}^0$-tree represented on the right:



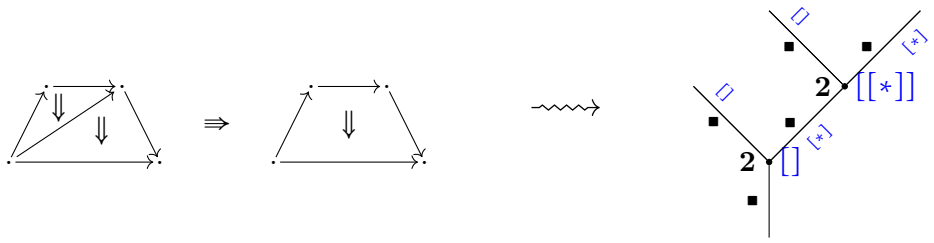
In the unnamed approach for opetopes presented in this section, $\mathbf{3}$ will be encoded as a mapping from its set of node addresses $\mathbf{3}^\bullet = \{[], [*], [**]\}$ to the set of 1-opetopes $\mathbb{O}_1$ as follows:

$$\mathbf{3} \rightsquigarrow \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases}$$

The 1-opetope $\blacksquare$ can recursively be encoded by $\{* \leftarrow \blacklozenge$, which gives a complete expression of **3**:

$$\begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \\ [**] \leftarrow \{* \leftarrow \blacklozenge \end{cases}$$

This example will be treated in depth in example 4.36. In a similar manner, the opetope ω on the left, whose tree is given in the middle, can be encoded as on the right:



$$\rightsquigarrow \quad \begin{cases} [] \leftarrow \mathbf{2} \\ [[*]] \leftarrow \mathbf{2} \end{cases}$$

Similarly to **3**, the opetope **2** can be expressed by $\begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$, and recall that $\blacksquare$ can be expressed by $\{* \leftarrow \blacklozenge$. We thus have a complete encoding of ω as:

$$\omega \quad \rightsquigarrow \quad \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \end{cases}$$

This example is fully treated in example 4.37.

4.1.1 Preopetopes

Definition 4.1 (Preopetope)**.** The sets $\mathbb{P}_n$ of *n-preopetopes* are defined by the following grammar:

$$\begin{aligned} \mathbb{P}_0 &::= \blacklozenge \\ \mathbb{P}_n &::= \begin{cases} \mathbb{A}_{n-1} \leftarrow \mathbb{P}_{n-1} \\ \vdots \\ \mathbb{A}_{n-1} \leftarrow \mathbb{P}_{n-1} \end{cases} && n \geq 1 && (4.2) \\ &\mid \quad \{\!\{\mathbb{P}_{n-2} && n \geq 2 && (4.3) \end{aligned}$$

where the set $\mathbb{A}_n$ of n-addresses is defined in definition 2.38. In line (4.2), we require further that there is at least one $(n-1)$-address, and that all addresses are distinct.

An n-preopetope $\mathbf{p}$ is *degenerate* if it is of the form of line (4.3), it is *non-degenerate* otherwise. We write $\dim \mathbf{p} := n$ for its *dimension*.

Convention 4.4. An n-preopetope as in equation (4.2) is considered as a *set* of expressions $\mathbb{A}_{n-1} \leftarrow \mathbb{P}_{n-1}$ rather than a list. For instance, the following two n-preopetopes are equal

$$\begin{cases} [p_1] \leftarrow \mathbf{p}_1 \\ [p_2] \leftarrow \mathbf{p}_2 \end{cases} = \begin{cases} [p_2] \leftarrow \mathbf{p}_2 \\ [p_1] \leftarrow \mathbf{p}_1 \end{cases}$$

for any distinct $(n-1)$-addresses $[p_1], [p_2] \in \mathbb{A}_{n-1}$, and any $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{P}_{n-1}$.

Example 4.5. (1) There is a unique 1-preopetope $\{* \leftarrow \blacklozenge$, which we simply write $\blacksquare$.

(2) The following are examples of a 2 and 3-preopetope, respectively:

$$\begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [* * * * *] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \qquad \begin{cases} [[*]] \leftarrow \{\{\blacklozenge \\ [[**][*][]] \leftarrow \{[] \leftarrow \{* \leftarrow \blacklozenge \end{cases}$$

We will see that the first does not correspond to an actual opetope, as it is impossible for a 2-opetope to only contain addresses [] and $[* * * * *]$ (it would at least need addresses $[*]$, $[**]$, $[* * *]$, and $[* * * *]$). The second does not correspond to an opetope either, as it does not have a root node (corresponding to address []).

(3) The following is a 4-preopetope $\{\{\{\blacklozenge$. We will see that it corresponds to $\mathsf{I}_{\blacklozenge} \in \mathbb{O}_4$.

(4) The following is not a valid preopetope:

$$\begin{cases} [[*]] \leftarrow \blacklozenge \\ [[*][*]] \leftarrow \{* \leftarrow \blacklozenge \end{cases}$$

as $\blacklozenge$ and $\{* \leftarrow \blacklozenge$ do not have the same dimension.

Definition 4.6. If we have a non-degenerate n-preopetope of the form

$$\mathbf{p} = \begin{cases} [p_1] \leftarrow \mathbf{q}_1 \\ \vdots \\ [p_k] \leftarrow \mathbf{q}_k \end{cases} \tag{4.7}$$

we call $[p_1], \dots, [p_k] \in \mathbb{A}_{n-1}$ the *source addresses* of $\mathbf{p}$ (or just *sources*), write $\mathbf{p}^\bullet$ for the set of source addresses of $\mathbf{p}$, and $\mathsf{s}_{[p_i]}\,\mathbf{p} := \mathbf{q}_i$ for the $[p_i]$-*source of* $\mathbf{p}$.

Assume $n \geq 2$. A *leaf address* (or just *leaf*) of $\mathbf{p}$ is an $(n-1)$-address of the form $[p[q]]$ such that $[p] \in \mathbf{p}^\bullet$, $[q] \in (\mathsf{s}_{[p]}\,\mathbf{p})^\bullet$, and such that for all $[r] \in \mathbf{p}^\bullet$, $[p[q]] \notin [r]$. In other words, $[p[q]]$ is not a prefix of any node address of $\mathbf{p}$, and in particular $[p[q]] \notin \mathbf{p}^\bullet$. We write $\mathbf{p}^! \subseteq \mathbb{A}_{n-1}$ for the set of leaf addresses of $\mathbf{p}$. By convention, if $\mathbf{p}$ is degenerate, then $\mathbf{p}^\bullet := \varnothing$ and $\mathbf{p}^! := \{[]\}$. Further, $\blacklozenge^\bullet = \blacklozenge^! := \varnothing$.

Example 4.8. Consider the following preopetopes

$$\mathbf{p} := \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \end{cases} \qquad \mathbf{q} := \begin{cases} [] \leftarrow \{[] \leftarrow \{* \leftarrow \blacklozenge \\ [[]] \leftarrow \{\{\blacklozenge \end{cases}$$

Then $\mathbf{p}^\bullet = \{[], [[*]]\}$, $\mathbf{p}^! = \{[[]], [[*][]], [[*][*]]\}$, $\mathbf{q}^\bullet = \{[], [[]]\}$, and $\mathbf{q}^! = \varnothing$.

Definition 4.9 (Corolla grafting)**.** Let $n \geq 1$, $\mathbf{p} \in \mathbb{P}_n$ be as in equation (4.7), and $\mathbf{q} \in \mathbb{P}_{n-1}$. For $[l] \in \mathbf{p}^!$ a leaf address of $\mathbf{p}$ (so in particular $[l] \notin \mathbf{p}^\bullet$), write

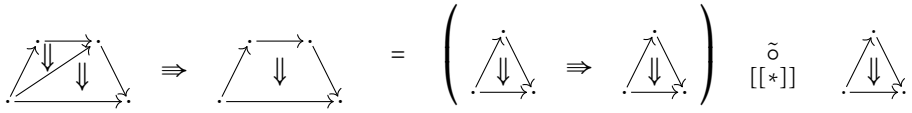
$$\mathbf{p} \underset{[l]}{\tilde{\circ}} \mathbf{q} := \begin{cases} [p_1] \leftarrow \mathbf{q}_1 \\ \vdots \\ [p_k] \leftarrow \mathbf{q}_k \\ [l] \leftarrow \mathbf{q} \end{cases}$$

and call $\mathbf{p} \,\tilde{\circ}_{[l]}\, \mathbf{q}$ the *corolla grafting* of $\mathbf{q}$ on $\mathbf{p}$ at address $[l]$. By convention, this operation is associative on the right.

Example 4.10. We have

$$\begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \end{cases} = \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \\ \qquad\qquad \tilde{\circ}_{[[*]]} \\ [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases}$$

which, together with the introduction of this chapter, means that graphically,



Remark 4.11. The denomination "corolla grafting" is motivated by the fact that $\mathbf{p}$ and $\mathbf{q}$ do not have the same dimension, and thus $\mathbf{q}$ needs to be made into a n-dimensional corolla first in order to be grafted in the sense of definition 2.13. Much like proposition 2.17, any preopetope can be obtained by iterated corolla grafting as follows.

Lemma 4.12. *Let $n \geq 1$, $\mathbf{p} \in \mathbb{P}_n$ be as in equation (4.7), and assume that whenever $1 \leq i < j \leq k$, we have either that $[p_i] \sqsubseteq [p_j]$ (definition 2.41), or $[p_i]$ and $[p_j]$ are $\sqsubseteq$-incomparable (in particular, this condition is satisfied if the $[p_i]$'s are lexicographically sorted). Then*

$$\mathbf{p} = \left(\cdots \left(\{[p_1] \leftarrow \mathbf{q}_1 \right) \tilde{\circ}_{[p_2]} \mathbf{q}_2 \cdots \right) \tilde{\circ}_{[p_k]} \mathbf{q}_k.$$

Proof. The condition on the sequence $[p_1], \ldots, [p_k]$ guarantees that the successive corolla graftings are well-defined, i.e. that for $1 \leq i < k$ and

$$\mathbf{p}_i \coloneqq \left(\cdots \left(\{[p_1] \leftarrow \mathbf{q}_1 \right) \tilde{\circ}_{[p_2]} \mathbf{q}_2 \cdots \right) \tilde{\circ}_{[p_i]} \mathbf{q}_i = \begin{cases} [p_1] \leftarrow \mathbf{q}_1 \\ \vdots \\ [p_i] \leftarrow \mathbf{q}_i \end{cases}$$

we have $[p_{i+1}] \in \mathbf{p}_i^!$. □

Lemma 4.13. *Let $n \geq 1$, $\mathbf{p} \in \mathbb{P}_n$ be as in equation (4.7), $\mathbf{q} \in \mathbb{P}_{n-1}$, and $[l] \in \mathbf{p}^!$. Then*

$$\left(\mathbf{p} \tilde{\circ}_{[l]} \mathbf{q} \right)^! = \mathbf{p}^! - \{[l]\} + \{[l[q]] \mid [q] \in \mathbf{q}^\bullet\}.$$

Proof. Easy verifications. □

4.1.2 Inference rules We now introduce a typing system for preopetopes in order to characterize those corresponding to opetopes, which is formally shown in theorem 4.34. We will deal with sequents of the following form.

Definition 4.14 (Sequent)**.** A *sequent* is an expression of the form

$$\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t},$$

where $\mathbf{p} \in \mathbb{P}_n$ for some $n \geq 0$, $\mathbf{t} \in \mathbb{P}_{n-1}$, and the *context* Γ is a finite set of pairs consisting of addresses $[l] \in \mathbf{p}^!$ and $[q] \in \mathbf{t}^\bullet$, denoted by $\frac{[l]}{[q]}$, such that a given $[q] \in \mathbf{t}^\bullet$ occurs at most once as a denominator. The preopetope $\mathbf{p}$ is the real object of interest as we will see in subsequent results. We may think of $\mathbf{t}$ as the "target" of $\mathbf{p}$, while Γ establishes a bijection between the leaves of $\mathbf{p}$ and the nodes of its target, playing the role of the readdressing map $\wp$ of definition 2.27.

Example 4.15. The following is a sequent:

$$\frac{[[]]}{[]}, \frac{[[*][]]}{[*]}, \frac{[[*][*]]}{[**]} \vdash \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases} \end{cases} \longrightarrow \begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \\ [**] \leftarrow \{* \leftarrow \blacklozenge \end{cases}$$

As we will see in example 4.37, it describes the following 3-opetope:

The operation of *substitution* (definitions 2.19 and 3.21), which consists in replacing a node by a pasting diagram in an opetope, can be defined as follows in our formalism.

Definition 4.16 (Substitution)**.** Let $\mathbf{t}, \mathbf{q} \in \mathbb{P}_n$, $\Upsilon \vdash \mathbf{q} \longrightarrow \mathbf{u}$ be a sequent. Write $\mathbf{t}$ as

$$\mathbf{t} = \begin{cases} [t_1] \leftarrow \mathbf{w}_1 \\ \vdots \\ [t_l] \leftarrow \mathbf{w}_l \end{cases}$$

For $[t_i] \in \mathbf{t}^\bullet$, we define $\mathbf{t} \,\square_{[t_i]}\, \mathbf{q}$, the *substitution* by $\mathbf{q}$ in $\mathbf{t}$ at $[t_i]$, as follows:

(1) if $l = 1$ and $\mathbf{q}$ is degenerate, then $\mathbf{t} \,\square_{[t_1]}\, \mathbf{q} := \mathbf{q}$;

(2) if $l \geq 2$ and $\mathbf{q}$ is degenerate, then

$$\mathbf{t} \underset{[t_i]}{\square} \mathbf{q} := \begin{cases} \rho[t_1] \leftarrow \mathbf{w}_1 \\ \vdots \\ \rho[t_{i-1}] \leftarrow \mathbf{w}_{i-1} \\ \rho[t_{i+1}] \leftarrow \mathbf{w}_{i+1} \\ \vdots \\ \rho[t_l] \leftarrow \mathbf{w}_l \end{cases} \quad \text{where} \quad \rho[t_j] := \begin{cases} [t_i r] & \text{if } [t_j] = [t_i[]r], \\ [t_j] & \text{otherwise.} \end{cases}$$

(3) if $l \geq 2$, and $\mathbf{q}$ is not degenerate, write it as

$$\mathbf{q} = \begin{cases} [q_1] \leftarrow \mathbf{v}_1 \\ \vdots \\ [q_k] \leftarrow \mathbf{v}_k \end{cases}$$

and define

$$\mathbf{t} \underset{[t_i]}{\square} \mathbf{q} := \begin{cases} \rho[t_1] \leftarrow \mathbf{w}_1 \\ \vdots \\ \rho[t_{i-1}] \leftarrow \mathbf{w}_{i-1} \\ [t_i q_1] \leftarrow \mathbf{v}_1 \\ \vdots \\ [t_i q_k] \leftarrow \mathbf{v}_k \\ \rho[t_{i+1}] \leftarrow \mathbf{w}_{i+1} \\ \vdots \\ \rho[t_l] \leftarrow \mathbf{w}_l \end{cases} \quad \text{where} \quad \rho[t_j] := \begin{cases} [t_i a r] & \text{if } [t_j] = [t_i[b]r] \\ & \text{for some } \frac{[a]}{[b]} \in \Upsilon, \\ [t_j] & \text{otherwise.} \end{cases}$$

This operation relies on the context Υ, which we leave implicit. By convention, $\square$ is associative on the left.

We refer to section 4.3 for examples of applications of this construction. We now state the inference rules of our unnamed system Opt$^?$ in definition 4.17.

Definition 4.17 (The Opt$^?$ system)**.** *Introduction of points.*

$$\frac{}{\vdash \blacklozenge \longrightarrow \varnothing}\ \texttt{point}$$

Introduction of degeneracies.

$$\frac{\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}}{\frac{[]}{[]} \vdash \{\!\{ \mathbf{p} \longrightarrow \{ [] \leftarrow \mathbf{p}}\ \texttt{degen}$$

Note that $\dim(\{\!\{ \mathbf{p}) = 2 + \dim \mathbf{p}$, and that $\dim(\{ [] \leftarrow \mathbf{p}) = 1 + \dim \mathbf{p}$.

Shift to the next dimension. Write $\mathbf{p}^\bullet = \{[p_1], \dots, [p_k]\}$.

$$\frac{\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}}{\frac{[[p_1]]}{[p_1]}, \dots, \frac{[[p_k]]}{[p_k]} \vdash \{ [] \leftarrow \mathbf{p} \longrightarrow \mathbf{p}}\ \texttt{shift}$$

As in the previous rule, $\dim(\{ [] \leftarrow \mathbf{p}) = 1 + \dim \mathbf{p}$.

Grafting. Assume $\dim \mathbf{p} = n \geq 2$, $[p[q]] \in \mathbf{p}^!$, $\dim \mathbf{q} = n - 1$, write $\mathbf{u} := \mathsf{s}_{[q]}\, \mathsf{s}_{[p]}\, \mathbf{p}$ and $\mathbf{q}^\bullet = \{[s_1], \dots, [s_l]\}$.

$$\frac{\Gamma, \frac{[p[q]]}{[r]} \vdash \mathbf{p} \longrightarrow \mathbf{t} \qquad \Upsilon \vdash \mathbf{q} \longrightarrow \mathbf{u}}{\Gamma', \frac{[p[q][s_1]]}{[rs_1]}, \dots, \frac{[p[q][s_l]]}{[rs_l]} \vdash \mathbf{p} \underset{[p[q]]}{\tilde{\circ}} \mathbf{q} \longrightarrow \mathbf{t} \underset{[r]}{\square} \mathbf{q}}\ \texttt{graft}$$

where Γ' is given by pairs of the form

(1) $\frac{[a]}{[rxr']}$, where $\frac{[a]}{[r[y]r']} \in \Gamma$ and $\frac{[x]}{[y]} \in \Upsilon$,

(2) $\frac{[a]}{[b]}$, where $\frac{[a]}{[b]} \in \Gamma$ is not as above (i.e. $[b]$ not of the form $[r[y]r']$ for some $\frac{[x]}{[y]} \in \Upsilon$).

In large derivation trees, we will sometimes refer to this rule as `graft-`$[p[q]]$ for clarity, or simply as $[p[q]]$ in order to make notations lighter.

Remark 4.18. Let us explain the transformation of context defined in rule `graft` in definition 4.17. Take a derivable sequent in Opt$^?$, say

$$\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t},$$

with $\mathbf{p} \in \mathbb{P}_n$. It will be proved in lemma 4.20 that Γ exhibits a bijection between $\mathbf{p}^!$ and $\mathbf{t}^\bullet$. Further, in theorem 4.34, we will see that $\mathbf{p}$ corresponds uniquely to an n-opetope $\omega = [\![\mathbf{p}]\!]$, that $\mathtt{t}\omega = [\![\mathbf{t}]\!]$, and that Γ corresponds to the readdressing function $\wp_\omega : \omega^! \longrightarrow (\mathtt{t}\omega)^\bullet$ or definition 2.27.

But where is the readdressing map $\wp_\omega$ implemented in Opt$^!$? Applying theorem 3.64, we know that ω corresponds to a unique sequent (modulo α-equivalence), say

$$E \triangleright \Upsilon \vdash x : s_1 \mapsto s_2 \mapsto \cdots \mapsto \varnothing$$

where $x \in \mathbb{V}_n$. More precisely, considered as a tree, ω is encoded by the term s_1, and by proposition 3.53, $\mathtt{t}\omega$ is encoded by s_2. In lemma 3.51, we show that $\wp_\omega$ exhibits a bijection

$$\{\&_{s_1} b \mid b \in s_2^\bullet\} \xrightarrow{\wp_\omega} \{\&_{s_2} b \mid b \in s_2^\bullet\}.$$

Say that a *node* of the term s_2 is a $(n-2)$-variable $b \in s_2^\bullet$, while a *leaf* of s_1 is a variable that can be used for grafting (see rule `graft` in definition 3.23), i.e. also a $(n-2)$-variable $b \in s_2^\bullet$. Then the left hand side can be considered as the set of leaf addresses of s_1, while the right hand side is its set of node addresses of s_2, and $\wp_\omega$ maps the address of $b \in \mathbb{V}_{n-2}$ as a leaf of s_1 to the address of b as a node of s_2. But in Opt$^!$, the function $\wp_\omega$ is unnecessary, as this correspondence is already established by the name of the variables.

In Opt$^?$ however, such bookkeeping is necessary since there are no names, and Γ is designed to precisely be the desired correspondence.

We now prove basic properties of derivable sequents in Opt$^?$. In proof trees, we may sometimes omit irrelevant information. For instance, if contexts and targets are not important, the `shift` rule may be written as

$$\frac{\mathbf{p}}{\{[] \leftarrow \mathbf{p}}\ \texttt{shift}$$

Lemma 4.19. *If* $\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}$ *is a derivable sequent, then* $\dim \mathbf{p} = 1 + \dim \mathbf{t}$.

Proof. Easy induction on proof trees. □

Lemma 4.20. *Let* $\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}$ *be a derivable sequent with* $\dim \mathbf{p} \geq 2$. *Then* Γ *establishes a bijection between* $\mathbf{p}^!$ *and* $\mathbf{t}^\bullet$ *(i.e. as a set of pairs,* Γ *is the graph of a bijective function).*

Proof. The fact that Γ is a relation from $\mathbf{p}^!$ to $\mathbf{t}^\bullet$ (i.e. that whenever $\frac{[a]}{[b]} \in \Gamma$ we have $[a] \in \mathbf{p}^!$ and $[b] \in \mathbf{t}^\bullet$) is clear from the inference rules. It is also clear that Γ is a function (i.e. that whenever $\frac{[a]}{[b]}, \frac{[a']}{[b']} \in \Gamma$ if $[b] \neq [b']$, then $[a] \neq [a']$). Finally, the fact that it is a bijection is clear in the case of `degen` and `shift`, and follows from lemma 4.13 in the case of `graft`. □

Lemma 4.21. *Let* $\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}$ *be a derivable sequent with* $\dim \mathbf{p} \geq 2$ *non-degenerate. For* $\frac{[p[q]]}{[r]} \in \Gamma$, *we have* $\mathsf{s}_{[q]}\, \mathsf{s}_{[p]}\, \mathbf{p} = \mathsf{s}_{[r]}\, \mathbf{t}$.

Proof. The sequent necessarily follows from an instance of `shift` or `graft`. The result is clear for the former, and follows from inspection for the latter. □

Proposition 4.22. *If* $\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}$ *is derivable, then so is* $\mathbf{t}$, *i.e. there exists a derivable sequent of the form* $\Upsilon \vdash \mathbf{t} \longrightarrow \mathbf{u}$.

Proof. The only non obvious case is (as always) `graft`, where we have to show that $\mathbf{t} \,\square_{[r]}\, \mathbf{q}$ is derivable. Since the sequent $\Gamma, \frac{[p[q]]}{[r]} \vdash \mathbf{p} \longrightarrow \mathbf{t}$ has a nonempty context, $\mathbf{p}$ and $\mathbf{t}$ are non-degenerate. Write $\mathbf{t}$ and $\mathbf{q}$ as in definition 4.16. Up to reindexing, we may assume that $\rho[t_j] = [t_j]$ if and only if $j < i$. Assume moreover that the sequences $[t_1], \ldots, [t_{i-1}]$ and $[t_{i+1}], \ldots, [t_l]$ are both lexicographically sorted. In particular, $[t_1] = []$. For $j > i$ write $[t_j] = [t_i[b_j]x_j]$ and $\wp[t_j] = [t_i a_j x_j]$, so that $\Upsilon = \left\{ \frac{[a_j]}{[b_j]} \mid i < j \leq l \right\}$. Then the proof tree of $\mathbf{t} \,\square_{[t_i]}\, \mathbf{q}$ is sketched as follows.

(1) If $[t_i] = []$, then necessarily $i = 1$, and $\mathbf{t} \,\square_{[t_i]}\, \mathbf{q}$ can be derived as

$$\dfrac{\dfrac{\dfrac{\vdots \quad \vdots}{\mathbf{q} \quad \mathbf{v}_2}}{\mathbf{q} \mathbin{\tilde{\circ}_{[a_2x_2]}} \mathbf{v}_2}\,\texttt{graft-}[a_2x_2] \quad \vdots\ \mathbf{v}_3}{\left(\mathbf{q} \mathbin{\tilde{\circ}_{[a_2x_2]}} \mathbf{v}_2\right) \mathbin{\tilde{\circ}_{[a_3x_3]}} \mathbf{v}_3}\,\texttt{graft-}[a_3x_3]$$

$$\vdots$$

$$\dfrac{\left(\cdots \left(\mathbf{q} \mathbin{\tilde{\circ}_{[a_2x_2]}} \mathbf{v}_2\right) \cdots\right) \mathbin{\tilde{\circ}_{[a_{l-1}x_{l-1}]}} \mathbf{v}_{l-1} \qquad \vdots\ \mathbf{v}_l}{\left(\cdots \left(\mathbf{q} \mathbin{\tilde{\circ}_{[a_2x_2]}} \mathbf{v}_2\right) \cdots\right) \mathbin{\tilde{\circ}_{[a_lx_l]}} \mathbf{v}_l}\,\texttt{graft-}[a_lx_l]$$

and by definition, $\left(\cdots \left(\mathbf{q} \mathbin{\tilde{\circ}_{[a_2x_2]}} \mathbf{v}_2\right) \cdots\right) \mathbin{\tilde{\circ}_{[a_lx_l]}} \mathbf{v}_l = \mathbf{t} \mathbin{\square_{[t_i]}} \mathbf{q}$.

(2) If $[t_i] \neq []$, then necessarily $i > 1$, and the process goes similarly. We first derive

$$\begin{cases} [t_1] \leftarrow \mathbf{v}_1 \\ \vdots \\ [t_{i-1}] \leftarrow \mathbf{v}_{i-1} \end{cases}$$

then graft the sources of $\mathbf{q}$, and lastly graft the remaining $\mathbf{v}_j$'s, where $j > i$. □

Proposition 4.23. *If* $\Gamma_1 \vdash \mathbf{p} \longrightarrow \mathbf{t}_1$ *and* $\Gamma_2 \vdash \mathbf{p} \longrightarrow \mathbf{t}_2$ *are two derivable sequents, then* $\Gamma_1 = \Gamma_2$ *(as sets) and* $\mathbf{t}_1 = \mathbf{t}_2$.

Proof. By inspection of the rules, Γ_i is completely determined by $\mathbf{p}^!$, thus $\Gamma_1 = \Gamma_2$. By lemma 4.20, Γ_i exhibits a bijection between $\mathbf{p}^!$ and $\mathbf{t}_i^\bullet$, and in particular, $\mathbf{t}_1^\bullet = \mathbf{t}_2^\bullet$. By lemma 4.21, for any $[p] \in \mathbf{t}_1^\bullet = \mathbf{t}_2^\bullet$, we have $\mathsf{s}_{[p]}\,\mathbf{t}_1 = \mathsf{s}_{[p]}\,\mathbf{t}_2$. Therefore, $\mathbf{t}_1 = \mathbf{t}_2$. □

Notation 4.24. We denote by $\mathbb{P}_n^\checkmark$ the set of derivable n-preopetopes, i.e. those $\mathbf{p}$ such that there exists a derivable sequent of the form $\Gamma \vdash \mathbf{p} \longrightarrow \mathbf{t}$. By proposition 4.23, this sequent is uniquely determined by $\mathbf{p}$ (it can in fact be *computed* from $\mathbf{p}$), so let $\mathtt{t}\,\mathbf{p} := \mathbf{t}$ be the *target* of $\mathbf{p}$, and $\wp_{\mathbf{p}} : \mathbf{p}^! \longrightarrow \mathbf{t}^\bullet$ be the bijection described by Γ. As such, the sequent around a derivable opetope $\mathbf{p}$ can be reconstructed as $\wp_{\mathbf{p}} \vdash \mathbf{p} \longrightarrow \mathtt{t}\,\mathbf{p}$.

Remark 4.25. Our syntax is closely related to the one given for multitopes [11, section 3], called here Hmp. Briefly, in Hmp, the unique 0 and 1-opetopes are respectively denoted ★ and # and, given an n-opetope $\mathbf{p}$, the notation $[\mathbf{p}]$ (resp. $\ulcorner \mathbf{p} \urcorner$) is used for the corresponding degenerate (resp. shifted) $(n+2)$- (resp. $(n+1)$-) opetope. The nodes of an opetope come equipped with a canonical order (just as in our system we could require preopetopes to be always sorted according to the lexicographical order $\preceq$). In Hmp, an inductive definition of opetopes is given, in the same spirit as our sequent calculus: in particular, typing conditions involving targets when grafting opetopes (grafting is simply application in Hmp) are involved. However, no explicit definition at the level of the syntax is given for computing targets (the description given resorts to multicategorical composition).

4.2 Equivalence with polynomial opetopes We now establish a series of definitions and results to show theorem 4.34, stating that the elements of $\mathbb{P}_n^\checkmark$ are in bijective correspondence with the set $\mathbb{O}_n$ of polynomial n-opetopes. Just as in the named case (section 3.2), we need to state a couple of definitions and results in one big induction whose scope goes all the way *from definition 4.26 to theorem 4.34 (included).*

Definition 4.26 (Unnamed coding)**.** Define the *unnamed coding function* $C^? : \mathbb{O}_n \longrightarrow \mathbb{P}_n$ by induction on $n \in \mathbb{N}$. If $n = 0, 1$, then $\mathbb{O}_n$ and $\mathbb{P}_n$ are singletons, so $C^?$ is trivially defined:

$$C^?(\blacklozenge) := \blacklozenge, \qquad C^?(\blacksquare) := \blacksquare = \{* \leftarrow \blacklozenge$$

Assume $n \geq 2$, that $C^?$ is defined for all $k < n$, and take $\omega \in \mathbb{O}_n$.

(1) If ω is degenerate, say $\omega = \mathsf{I}_\phi$ for some $\phi \in \mathbb{O}_{n-2}$, then

$$C^?\left(\mathsf{I}_\phi\right) \coloneqq \{\!\{C^?\left(\phi\right) .$$

(2) If ω is an endotope, say $\omega = \mathsf{Y}_\psi$ for some $\psi \in \mathbb{O}_{n-1}$, then

$$C^?\left(\mathsf{Y}_\psi\right) \coloneqq \left\{[] \leftarrow C^?\left(\psi\right) .$$

(3) Otherwise, decompose ω as $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for $\nu \in \mathbb{O}_n$, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$, and let

$$C^?\left(\omega\right) \coloneqq C^?\left(\nu\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right).$$

This grafting is well-defined by lemma 4.28 applied inductively to ν.

Proposition 4.27. *Let $n \geq 2$ and $\omega \in \mathbb{O}_n$ have at least three nodes. Then the preopetope $C^?\left(\omega\right)$ does not depend on the decomposition of ω in corollas.*

Proof. Akin to proposition 3.62, it is enough to check that for $\nu \in \mathbb{O}_n$ non-degenerate, two different leaf addresses $[l], [l'] \in \nu^!$ (which are necessarily $\sqsubseteq$-incomparable), and $\psi, \psi' \in \mathbb{O}_{n-1}$ such that $\mathsf{t}\psi = \mathsf{e}_{[l]}\nu$ and $\mathsf{t}\psi' = \mathsf{e}_{[l']}\nu$, we have

$$C^?\left(\left(\nu \underset{[l]}{\circ} \mathsf{Y}_\psi\right) \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) = C^?\left(\left(\nu \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) \underset{[l]}{\circ} \mathsf{Y}_\psi\right).$$

To unclutter notations, write $C^?\left(\nu\right) = \left\{\vdots\right.$. We have

$$\begin{aligned}
&C^?\left(\left(\nu \underset{[l]}{\circ} \mathsf{Y}_\psi\right) \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) \\
&= C^?\left(\nu \underset{[l]}{\circ} \mathsf{Y}_\psi\right) \underset{[l']}{\tilde{\circ}} C^?\left(\psi'\right) && \text{by definition} \\
&= \left(C^?\left(\nu\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right)\right) \underset{[l']}{\tilde{\circ}} C^?\left(\psi'\right) && \text{by definition} \\
&= \left(\left\{\begin{array}{l}\vdots\\ [l] \leftarrow C^?\left(\psi\right)\end{array}\right.\right) \underset{[l']}{\tilde{\circ}} C^?\left(\psi'\right) && \text{see definition 4.9} \\
&= \left\{\begin{array}{l}\vdots\\ [l] \leftarrow C^?\left(\psi\right)\\ [l'] \leftarrow C^?\left(\psi'\right)\end{array}\right. && \text{see definition 4.9} \\
&= \left\{\begin{array}{l}\vdots\\ [l'] \leftarrow C^?\left(\psi'\right)\\ [l] \leftarrow C^?\left(\psi\right)\end{array}\right. && \text{see convention 4.4} \\
&= \left(\left\{\begin{array}{l}\vdots\\ [l'] \leftarrow C^?\left(\psi'\right)\end{array}\right.\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right) && \text{since } [l] \not\sqsubseteq [l'] \\
&= \left(C^?\left(\nu\right) \underset{[l']}{\tilde{\circ}} C^?\left(\psi'\right)\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right) \\
&= C^?\left(\left(\nu \underset{[l']}{\circ} \mathsf{Y}_{\psi'}\right) \underset{[l]}{\circ} \mathsf{Y}_\psi\right).
\end{aligned}$$

□

We now establish a series of results to prove proposition 4.31 stating that $C^?(\omega)$ is always a derivable preopetope.

Lemma 4.28. *For $\omega \in \mathbb{O}_n$, we have $\omega^\bullet = C^?(\omega)^\bullet$, and $\omega^! = C^?(\omega)^!$.*

Proof. We proceed by induction.

(1) If $n \leq 1$, then the claims trivially hold.

(2) Assume $\omega = \mathsf{I}_\phi$, for some $\phi \in \mathbb{O}_{n-2}$. Then $\omega^\bullet = \varnothing = \left(\{\!\{C^?(\phi)\}\!\}\right)^\bullet$. For leaves, $\omega^! = \{[]\} = \left(\{\!\{C^?(\phi)\}\!\}\right)^! = C^?(\omega)^!$ (see definition 4.6).

(3) Assume $\omega = \mathsf{Y}_\psi$, for some $\psi \in \mathbb{O}_{n-1}$, so that we have $C^?(\omega) = \{[] \leftarrow C^?(\psi)\}$. Then $\omega^\bullet = \{[]\} = \left(\{[] \leftarrow C^?(\psi)\}\right)^\bullet = C^?(\omega)^\bullet$. By induction, $\psi^\bullet = C^?(\psi)^\bullet$, so the leaf addresses of ω and $\{[] \leftarrow C^?(\psi)$ are both of the form $[[q]]$, where $[q]$ ranges over $\psi^\bullet$, hence $\omega^! = C^?(\omega)^!$.

(4) Assume $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for some $\nu \in \mathbb{O}_n$, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$. Then, by induction,

$$\begin{aligned}
\omega^\bullet &= \nu^\bullet + \{[l]\} && \text{by definition}\\
&= C^?(\nu)^\bullet + \{[l]\} && \text{by induction}\\
&= \left(C^?(\nu) \underset{[l]}{\tilde{\circ}} C^?(\psi)\right)^\bullet && \text{see definition 4.9}\\
&= C^?(\omega)^\bullet && \text{see definition 4.26,}
\end{aligned}$$

and

$$\begin{aligned}
\omega^! &= \nu^! - \{[l]\} + \{[l[q]] \mid [q] \in \psi^\bullet\} && \text{by definition}\\
&= C^?(\nu)^! - \{[l]\} + \left\{[l[q]] \mid [q] \in C^?(\psi)^\bullet\right\} && \text{by induction}\\
&= \left(C^?(\nu) \underset{[l]}{\tilde{\circ}} C^?(\psi)\right)^! && \text{see definition 4.9}\\
&= C^?(\omega)^! && \text{see definition 4.26.}
\end{aligned}$$

□

Lemma 4.29. *For $\omega \in \mathbb{O}_n$ non-degenerate and $[p] \in \omega^\bullet$, we have $C^?\left(\mathsf{s}_{[p]}\,\omega\right) = \mathsf{s}_{[p]}\,C^?(\omega)$.*

Proof. We proceed by induction. Since $\omega^\bullet \neq \varnothing$, ω is either $\blacksquare$, an endotope, or a grafting.

(1) If $\omega = \blacksquare$, then $[p] = *$, and trivially, $C^?(\mathsf{s}_*\,\blacksquare) = C^?(\blacklozenge) = \blacklozenge = \mathsf{s}_*\{* \leftarrow \blacklozenge = \mathsf{s}_*\,C^?(\blacksquare)$.

(2) Assume that ω is an endotope, say $\omega = \mathsf{Y}_\psi$, for some $\psi \in \mathbb{O}_{n-1}$. Necessarily, $[p] = []$, and $C^?\left(\mathsf{s}_{[]}\,\omega\right) = C^?(\psi) = \mathsf{s}_{[]}\{[] \leftarrow C^?(\psi) = \mathsf{s}_{[]}\,C^?(\omega)$.

(3) Assume $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for some $\nu \in \mathbb{O}_n$, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$. Let $[p] \in \omega^\bullet$. If $[p] = [l]$, then

$$\begin{aligned}
C^?\left(\mathsf{s}_{[l]}\,\omega\right) &= C^?(\psi)\\
&= \mathsf{s}_{[l]}\left(C^?(\nu) \underset{[l]}{\tilde{\circ}} C^?(\psi)\right) && \text{see definition 4.9}\\
&= \mathsf{s}_{[l]}\,C^?(\omega) && \text{see definition 4.26.}
\end{aligned}$$

Otherwise, we have

$$\begin{aligned} C^?\left(\mathsf{s}_{[p]}\,\omega\right) &= C^?\left(\mathsf{s}_{[p]}\,\nu\right) \\ &= \mathsf{s}_{[p]}\,C^?\left(\nu\right) && \text{by induction} \\ &= \mathsf{s}_{[p]}\left(C^?\left(\nu\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right)\right) && \text{see definition 4.9} \\ &= \mathsf{s}_{[p]}\,C^?\left(\omega\right) && \text{see definition 4.26.} \end{aligned}$$

□

Lemma 4.30. *Let $\omega \in \mathbb{O}_n$, and assume that $C^?(\omega)$ is derivable. Then $C^?(\mathsf{t}\omega) = \mathsf{t}C^?(\omega)$, and $\wp_\omega = \wp_{C^?(\omega)}$ (see notation 4.24).*

Proof. We proceed by induction.

(1) Assume $\omega = \mathsf{I}_\phi$, for some $\phi \in \mathbb{O}_{n-2}$. Then

$$\begin{aligned} C^?\left(\mathsf{t}\omega\right) &= C^?\left(\mathsf{Y}_\phi\right) \\ &= \left\{[] \leftarrow C^?\left(\phi\right) && \text{see definition 4.26} \right. \\ &= \mathsf{t}\left\{\!\left\{C^?\left(\phi\right) \right.\right. && \text{see } \texttt{degen} \text{ rule} \\ &= \mathsf{t}C^?\left(\omega\right) && \text{see definition 4.26.} \end{aligned}$$

Since ω and $C^?(\omega)$ are both degenerate (as opetope and preopetope, respectively), $\wp_\omega$ and $\wp_{C^?(\omega)}$ both map $[] \in \omega^! = C^?(\omega)^!$ to $[] \in (\mathsf{t}\omega)^\bullet = \mathsf{t}C^?(\omega)^\bullet$ (see the `degen` rule).

(2) Assume $\omega = \mathsf{Y}_\psi$, for some $\psi \in \mathbb{O}_{n-1}$, so that we have $C^?(\omega) = \left\{[] \leftarrow C^?(\psi)\right.$. Then

$$\begin{aligned} C^?\left(\mathsf{t}\omega\right) &= C^?\left(\psi\right) \\ &= \mathsf{t}\left\{[] \leftarrow C^?\left(\psi\right)\right. && \text{see } \texttt{shift} \text{ rule} \\ &= \mathsf{t}C^?\left(\omega\right) && \text{see definition 4.26.} \end{aligned}$$

Moreover, we have $\omega^! = C^?(\omega)^! = \{[[p]] \mid [p] \in \psi^\bullet\}$, and by definition, $\wp_\omega[[p]] = [p] = \wp_{C^?(\omega)}[p]$ (see the `shift` rule).

(3) Assume $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for some $\nu \in \mathbb{O}_n$, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$. Then,

$$\begin{aligned} C^?\left(\mathsf{t}\omega\right) &= C^?\left((\mathsf{t}\nu) \underset{\wp_\nu[l]}{\square} \psi\right) && \text{by proposition 2.37} \\ &= C^?\left(\mathsf{t}\nu\right) \underset{\wp_{C^?(\nu)}[l]}{\square} C^?\left(\psi\right) && \text{by induction, } \wp_\nu = \wp_{C^?(\nu)} \\ &= \mathsf{t}C^?\left(\nu\right) \underset{\wp_{C^?(\nu)}[l]}{\square} C^?\left(\psi\right) && \text{by induction} \\ &= \mathsf{t}\left(C^?\left(\nu\right) \underset{[l]}{\tilde{\circ}} C^?\left(\psi\right)\right) && \text{see } \texttt{graft} \text{ rule} \\ &= \mathsf{t}C^?\left(\omega\right) \end{aligned}$$

The equality $\wp_\omega = \wp_{C^?(\omega)}$ follows by inspection of rule `graft`. □

Proposition 4.31. *If $\omega \in \mathbb{O}_n$, then $C^?(\omega)$ is a derivable n-preopetope.*

Proof. If $n \leq 1$, then the result is trivial, so assume that $n \geq 2$.

(1) If $\omega = \mathsf{l}_\phi$ for some $\phi \in \mathbb{O}_{n-2}$, then $C^?(\omega)$ is simply obtained by an instance of `degen`:

$$\frac{\begin{array}{c}\vdots\\ C^?(\phi)\end{array}}{C^?(\omega)}\,\texttt{degen}$$

(2) Likewise, if $\omega = \mathsf{Y}_\psi$ for some $\psi \in \mathbb{O}_{n-1}$, then $C^?(\omega)$ can be obtained using an instance of `shift`.

(3) Assume that $\omega = \nu \circ_{[l]} \mathsf{Y}_\psi$, for $\nu\mathbb{O}_n$ non-degenerate, $\psi \in \mathbb{O}_{n-1}$, and $[l] \in \nu^!$. By lemma 4.28, $\nu^! = C^?(\nu)^!$, thus $[l] \in C^?(\nu)^!$. Since ν is not degenerate, the leaf address $[l]$ can be decomposed as $[l] = [p[q]]$, where $[p] \in \nu^\bullet$ and $[q] \in (\mathsf{s}_{[p]}\,\nu)^\bullet$. We have

$$\begin{aligned}
\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,C^?(\nu) &= \mathsf{s}_{[q]}\,C^?\left(\mathsf{s}_{[p]}\,\nu\right) && \text{by lemma 4.28}\\
&= C^?\left(\mathsf{s}_{[q]}\,\mathsf{s}_{[p]}\,\nu\right) && \text{by lemma 4.28}\\
&= C^?\left(\mathsf{t}\,\psi\right) && \text{by assumption}\\
&= \mathsf{t}\,C^?(\psi) && \text{by lemma 4.30}
\end{aligned}$$

Finally, rule `graft` can be used to derive $C^?(\omega)$:

$$\frac{\begin{array}{cc}\vdots & \vdots\\ C^?(\nu) & C^?(\psi)\end{array}}{C^?(\omega)}\,\texttt{graft-}[l]$$

□

We finally prove that $C^?$ is a bijection by constructing its inverse.

Definition 4.32. Define the *polynomial coding function* $[\![-]\!] : \mathbb{P}_n^\checkmark \longrightarrow \mathbb{O}_n$ by induction on $n \in \mathbb{N}$. If $n = 0, 1$, then both sets are singletons, and $[\![-]\!]$ is trivially defined. Assume $n \geq 2$, and that $[\![-]\!]$ is defined for all $k < n$. We distinguish three cases.

(1) If $\mathbf{q} \in \mathbb{P}_{n-2}^\checkmark$, then $\left[\!\left[\{\!\{ \mathbf{q} \}\!\} \right]\!\right] := \mathsf{l}_{[\![\mathbf{q}]\!]}$.

(2) If $\mathbf{q} \in \mathbb{P}_{n-1}^\checkmark$, then $\left[\!\left[\{ [] \leftarrow \mathbf{q} \} \right]\!\right] := \mathsf{Y}_{[\![\mathbf{q}]\!]}$.

(3) If $\mathbf{p} \in \mathbb{P}_n^\checkmark$, $\mathbf{q} \in \mathbb{P}_{n-1}^\checkmark$, and $[l] \in \mathbf{p}^!$ are such that the corresponding instance of rule `graft` is well-defined, then let

$$\left[\!\left[\mathbf{p} \underset{[l]}{\tilde{\circ}} \mathbf{q} \right]\!\right] := [\![\mathbf{p}]\!] \underset{[l]}{\circ} \mathsf{Y}_{[\![\mathbf{q}]\!]},$$

which is well-defined by lemmas 4.28 to 4.30, and inductively applying theorem 4.34.

Lemma 4.33. *Let $n \geq 1$ and $\mathbf{p} \in \mathbb{P}_n^\checkmark$ have at least three node addresses. Then $[\![\mathbf{p}]\!]$ does not depend on the decomposition of $\mathbf{p}$ into corolla graftings.*

Proof. Akin to propositions 3.62 and 4.27, it is enough to check that for $\mathbf{p} \in \mathbb{P}_n^\checkmark$ non-degenerate, $[l], [l'] \in \mathbf{p}^!$ distinct leaf addresses (in particular, they are $\sqsubseteq$-incomparable), $\mathbf{q}, \mathbf{q}' \in \mathbb{P}_{n-1}^\checkmark$ such that $\mathsf{t}\,\mathbf{q} = \mathsf{e}_{[l]}\,\mathbf{p}$ and $\mathsf{t}\,\mathbf{q}' = \mathsf{e}_{[l']}\,\mathbf{p}$, we have

$$\left[\!\left[\left(\mathbf{p} \underset{[[l]]}{\tilde{\circ}} \mathbf{q} \right) \underset{[[l']]}{\tilde{\circ}} \mathbf{q}' \right]\!\right] = \left[\!\left[\left(\mathbf{p} \underset{[[l']]}{\tilde{\circ}} \mathbf{q}' \right) \underset{[[l]]}{\tilde{\circ}} \mathbf{q} \right]\!\right].$$

This is straightforward:

$$\begin{aligned}
\left[\!\!\left[\left(\mathbf{p} \underset{[[l]]}{\tilde{\circ}} \mathbf{q}\right) \underset{[[l']]}{\tilde{\circ}} \mathbf{q}'\right]\!\!\right] &= \left(\llbracket \mathbf{p} \rrbracket \underset{[[l]]}{\circ} \mathsf{Y}_{\llbracket \mathbf{q} \rrbracket}\right) \underset{[[l']]}{\circ} \mathsf{Y}_{\llbracket \mathbf{q}' \rrbracket} && \text{by definition} \\
&= \left(\llbracket \mathbf{p} \rrbracket \underset{[[l']]}{\circ} \mathsf{Y}_{\llbracket \mathbf{q}' \rrbracket}\right) \underset{[[l]]}{\circ} \mathsf{Y}_{\llbracket \mathbf{q} \rrbracket} && \text{since } [[l]] \notin [[l']] \\
&= \left[\!\!\left[\left(\mathbf{p} \underset{[[l']]}{\tilde{\circ}} \mathbf{q}'\right) \underset{[[l]]}{\tilde{\circ}} \mathbf{q}\right]\!\!\right] && \text{by definition.}
\end{aligned}$$

□

Theorem 4.34. *The functions $C^?$ and $\llbracket - \rrbracket$ are mutually inverse.*

Proof. It is straightforward to check that $\llbracket C^? (\omega) \rrbracket = \omega$ by induction on the dimension and the number of nodes of ω, and that $C^? (\llbracket \mathbf{p} \rrbracket) = \mathbf{p}$ by induction on the prooftree of $\mathbf{p}$. □

This concludes the inductive process started at definition 4.26.

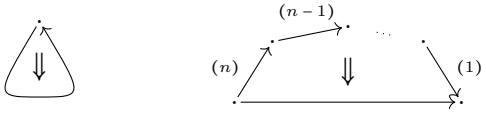
4.3 Examples

In this section, we give example derivations in system Opt$^?$.

Example 4.35 (The arrow)**.** The unique 1-opetope $\blacksquare = \{ * \leftarrow \blacklozenge$ is derived by

$$\dfrac{\dfrac{}{\vdash \blacklozenge \longrightarrow \varnothing}\,\texttt{point}}{\vdash \{ * \leftarrow \blacklozenge \longrightarrow \blacklozenge}\,\texttt{shift}$$

Example 4.36 (Opetopic integers)**.** The opetopic integer $\mathbf{n}$ (example 2.36) is represented on the left in the case $n = 0$, and on the right if $n \geq 1$:



The derivation of $\mathbf{0}$ is simply

$$\dfrac{\dfrac{}{\vdash \blacklozenge}\,\texttt{point}}{[] \vdash \{\!\{ \blacklozenge \longrightarrow \blacksquare}\,\texttt{degen}$$

whereas for $n \geq 1$, the opetope $\mathbf{n}$ is derived as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash \blacksquare \longrightarrow \blacklozenge}}{\dfrac{[*]}{*} \vdash \{[] \leftarrow \blacksquare \longrightarrow \blacksquare}\,\texttt{shift} \qquad \vdots \quad \vdash \blacksquare \longrightarrow \blacklozenge}{\dfrac{[**]}{*} \vdash \begin{cases}[] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare\end{cases} \longrightarrow \blacksquare}\,[*] \qquad \vdots \quad \vdash \blacksquare \longrightarrow \blacklozenge}{\dfrac{[***]}{*} \vdash \begin{cases}[] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare\end{cases} \longrightarrow \blacksquare}\,[**]}{\dfrac{[*^{n-1}]}{*} \vdash \begin{cases}[] \leftarrow \blacksquare \\ \vdots \\ [*^{n-2}] \leftarrow \blacksquare\end{cases} \longrightarrow \blacksquare \qquad \vdots \quad \vdash \blacksquare \longrightarrow \blacklozenge}\,[*^{n-1}]}{\dfrac{[*^{n}]}{*} \vdash \begin{cases}[] \leftarrow \blacksquare \\ \vdots \\ [*^{n-1}] \leftarrow \blacksquare\end{cases} \longrightarrow \blacksquare}$$

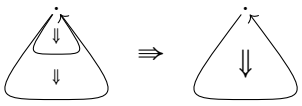
where there is a total of $n-1$ instances of the `graft` rule.

Example 4.37 (A classic)**.** The 3-opetope

can be derived as follows:

$$\dfrac{\dfrac{\dfrac{\vdots}{\dfrac{[**]}{*} \vdash \mathbf{2} \longrightarrow \blacksquare}}{\dfrac{[[*]]}{[*]} \vdash \{[] \leftarrow \mathbf{2} \longrightarrow \mathbf{2}}\,\texttt{shift} \qquad \vdots \quad \dfrac{[**]}{*} \vdash \mathbf{2} \longrightarrow \blacksquare}{\dfrac{[[]]}{[]}, \dfrac{[[*][]]}{[*]}, \dfrac{[[*][*]]}{[**]} \vdash \begin{cases}[] \leftarrow \mathbf{2} \\ [[*]] \leftarrow \mathbf{2}\end{cases} \longrightarrow \mathbf{3}}\,\texttt{graft-}[[*]]$$

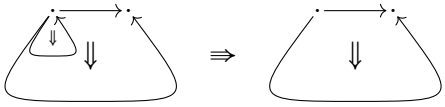
Example 4.38 (A degenerate case)**.** The 3-opetope



can be derived as follows:

$$\dfrac{\dfrac{\dfrac{\vdots}{\dfrac{[*]}{*} \vdash \mathbf{1} \longrightarrow \blacksquare}}{\dfrac{[[]]}{[]} \vdash \{[] \leftarrow \mathbf{1} \longrightarrow \mathbf{1}}\,\texttt{shift} \qquad \vdots \quad \dfrac{[]}{[]} \vdash \mathbf{0} \longrightarrow \blacksquare}{\vdash \begin{cases}[] \leftarrow \mathbf{1} \\ [[]] \leftarrow \mathbf{0}\end{cases} \longrightarrow \mathbf{0}}\,\texttt{graft-}[[]]$$

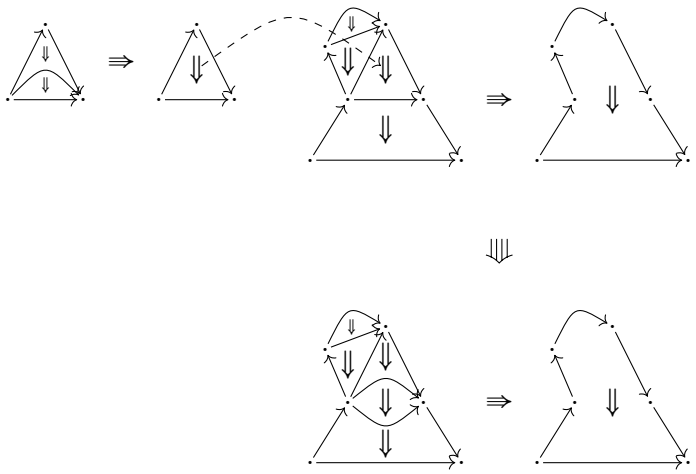
Example 4.39 (Another degenerate case)**.** The 3-opetope



can be derived as follows:

$$\dfrac{\dfrac{\dfrac{\vdots}{\frac{[**]}{*} \vdash \mathbf{2} \longrightarrow \blacksquare}}{\frac{[[]]}{[]}, \frac{[[*]]}{[*]} \vdash \{[] : \mathbf{2} \longrightarrow \mathbf{2}}\ \texttt{shift} \qquad \dfrac{\vdots}{\vdash \mathbf{0} \longrightarrow \blacksquare}}{\frac{[[]]}{[]} \vdash \begin{cases} [] \leftarrow \mathbf{2} \\ [[*]] \leftarrow \mathbf{0} \end{cases} \longrightarrow \mathbf{1}}\ \texttt{graft-}[[*]]$$

Example 4.40 (A 4-opetope)**.** The 4-opetope



is derived by

4.4 Deciding opetopes We now present in algorithm 4.1 the isOpetope function that, given a preopetope $\omega \in \mathbb{P}$, decides if $\omega \in \mathbb{P}^{\checkmark}$, as proved in proposition 4.41. This algorithm tries to deconstruct ω by finding the last rule instance of its potential proof tree, and recursively checking the validity of the premises. We emphasize that this algorithm is extremely inefficient.

Algorithm 4.1 Well formation algorithm

```
1:  procedure isOpetope(ω ∈ ℙ)                                  ▷ Returns a boolean
2:      if ω = ♦ then
3:          return true
4:      else if ω = {{φ then
5:          return isOpetope(φ)
6:      else
7:          while ω has an address of the form [p[q]] do
8:              if [p] ∉ ω• or not isOpetope(s_[p] ω) then
9:                  return false
10:             else if [q] ∉ (s_[p] ω)• or not isOpetope(s_[p[q]] ω) then
11:                 return false
12:             else if t s_[p] ω ≠ s_[p[q]] ω then
13:                 return false
14:             else
15:                 Remove address [p[q]] from ω
16:             end if
17:         end while
18:         if ω is of the form {[] ← ψ then
19:             return isOpetope(ψ)
20:         else
21:             return false
22:         end if
23:     end if
24: end procedure
```

Proposition 4.41. *For $\omega \in \mathbb{P}$, the execution* isOpetope(ω) *returns* ***true*** *if and only if $\omega \in \mathbb{P}^{\checkmark}$.*

Proof. This algorithm tries to deconstruct the potential proof tree of ω in system Opt$^?$:

(1) condition at line (2) removes an instance of the `point` rule;
(2) condition at line (4) removes an instance of `degen`;
(3) each iteration of the **while** loop at line (7) removes an instance of `graft`;
(4) finally the condition at line (18) removes an instance of `shift`.

If the algorithm encounters an expression that is not the conclusion of any instance of any rule of Opt$^?$, it returns **false**. Otherwise, if all branches of the proof tree lead to $\blacklozenge$, it returns **true**. □

This algorithm is implemented in method `UnnamedOpetope.ProofTree` of package `opetopy` [13]. It is worth noting that deciding opetopes can also be achieved by enumerating the proof trees of Opt$^?$. In fact, this enumeration scheme applies to all derivation systems presented in this work.

4.5 Python implementation The derivation system Opt$^?$ and its syntax are implemented in `opetopy.UnnamedOpetope` of [13]. In particular, the four derivation rules are represented by functions of the same name: `point`, `degen`, `shift`, and `graft`. As an example, we review the implementation of `shift` in figure 4.1.

Figure 4.1: Implementation of Opt$^?$'s `shift` rule in `opetopy.UnnamedOpetope.shift`

```
# This function takes a sequent (opetopy.UnnamedOpetope.Sequent) and returns
↪ a sequent. A sequent seq is structured as follows: for seq = Γ ⊢ s ⟶ t we
↪ have seq.context = Γ, seq.source = s, and seq.target = t.
def shift(seq: Sequent) -> Sequent:
    # We let n be the dimension of the preopetope s = seq.source
    n = seq.source.dimension
    # We construct a new context ctx dimension n + 1
    ctx = Context(n + 1)
    # We let ctx = { [a]/a | a ∈ s• }
    for a in seq.source.nodeAddresses():
        ctx += (a.shift(), a)
    # We return the sequent ctx ⊢ {[] ← s ⟶ s}
    return Sequent(
        ctx,
        Preopetope.fromDictOfPreopetopes({
            Address.epsilon(n): seq.source
        }),
        seq.source
    )
```

To construct proof trees, those rules are further abstracted in classes `Point`, `Degen`, `Shift`, as well as `Graft`. We review the implementation of some examples in figures 4.2 to 4.4.

Figure 4.2: Derivation of the arrow sequent (see example 4.35) using `opetopy.UnnamedOpetope`

```
from opetopy.UnnamedOpetope import *

# The arrow sequent is obtained by an application of the point rule followed
 ↪  by an application of the shift rule.
arrow = Shift(Point())
# Note that the function opetopy.UnnamedOpetope.Arrow can be used to
 ↪  concisely get the proof tree of ▪.
```

Figure 4.3: Derivation of some opetopic integers (see example 4.36) using `opetopy.UnnamedOpetope`

```
from opetopy.UnnamedOpetope import *

# The opetopic integer 0 is obtained by an instance of the point rule,
 ↪  followed by an application of the degen rule.
opetopic_integer_0 = Degen(Point())
# The opetopic integer 1 is obtained by applying rule shift to the arrow ▪ as
 ↪  defined in the previous figure.
opetopic_integer_1 = Shift(arrow)
# The opetopic integer 2 is defined by 2 = 1 ○̃[*] ▪. The address [*] is obtained
 ↪  with the convenient UnnamedOpetope.address function (as opposed to using
 ↪  the UnnamedOpetope.Address class).
opetopic_integer_2 = Graft(
    opetopic_integer_1,
    arrow,
    address(["*"]))
# Likewise, 3 = 2 ○̃[**] ▪.
opetopic_integer_3 = Graft(
    opetopic_integer_2,
    arrow,
    address(["*", "*"]))
# Note that the function opetopy.UnnamedOpetope.OpetopicInteger can be used
 ↪  to get the proof tree of an arbitrary opetopic integer.
```

Figure 4.4: Derivation of example 4.37 using `opetopy.UnnamedOpetope`

```
from opetopy.UnnamedOpetope import *

# Recall that in this example, the final opetope ω is defined by
↪  ω := ({[] ← 2} ∘̃_{[[*]]} 2.
example_classic = Graft(
    Shift(opetopic_integer_2),
    opetopic_integer_2,
    address([["*"]]))
```

4.6 The system for opetopic sets

Definition 4.42 (Pasting diagram)**.** Let $\mathbf{p} \in \mathbb{P}^{\checkmark}$. We define the notion of *pasting diagram* of shape $\mathbf{p}$:

(1) if $\mathbf{p} = \blacklozenge$, then the expression "$\blacklozenge$" is the only pasting diagram of shape $\blacklozenge$;
(2) if $\mathbf{p}$ is degenerate, then a pasting diagram of shape $\mathbf{p}$ is an expression of the form $\{\!\{x$, where x is a variable;
(3) otherwise, if $\{[p_1], \dots, [p_k]\} = \mathbf{p}^{\bullet}$, then a pasting diagram of shape $\mathbf{p}$ is an expression of the form

$$\begin{cases} [p_1] \leftarrow x_1 \\ \vdots \\ [p_k] \leftarrow x_k \end{cases}$$

where $x_1, \dots, x_k$ are variables.

If $\mathbf{P}$ is a pasting diagram of shape $\mathbf{p}$, then we write $\mathbf{P}^{\natural} = \mathbf{p}$. In the third case, we also write $\mathsf{s}_{[p_i]}\,\mathbf{P} = x_i$.

Definition 4.43 (Typing)**.** A *typing* is an expression of the form $x : \mathbf{P} \longrightarrow y$, where $\mathbf{P}$ is a pasting diagram, and where x and y are variables not occurring in $\mathbf{P}$. The *shape* of x is then simply the shape of $\mathbf{P}$, and we write $x^{\natural} = \mathbf{P}^{\natural}$. Let $\mathsf{t}\,x := y$, and if $[p] \in (x^{\natural})^{\bullet}$, let $\mathsf{s}_{[p]}\,x := \mathsf{s}_{[p]}\,\mathbf{P}$.

Definition 4.44 (Context)**.** A *context* Γ is an ordered list of typings of distinct variables.

In the OptSet$^?$ system, we rely on two types of judgment that can be understood as follows:
(1) "Γ", meaning that Γ is a context,
(2) "$\Gamma \vdash \mathbf{P}$", meaning that $\mathbf{P}$ is a pasting diagram in context Γ.

We now state the inference rules in definition 4.45.

Definition 4.45 (The OptSet$^?$ system)**.** *Introduction of points.*

$$\frac{\Gamma}{\Gamma, x : \blacklozenge}\ \texttt{point}$$

for x a fresh variable name.

Introduction of degenerate pasting diagrams.

$$\frac{\Gamma, x : T}{\Gamma, x : T \vdash \{\!\{ x} \; \texttt{degen}$$

The shape of this pasting diagram is $(\{\!\{ x)^\natural := \{\!\{ x^\natural$.

Introduction of non-degenerate pasting diagrams. If there exists $\mathbf{p} \in \mathbb{P}^\checkmark$ a non-degenerate opetope, say

$$\mathbf{p} = \begin{cases} [p_1] \leftarrow \psi_1 \\ \vdots \\ [p_k] \leftarrow \psi_k \end{cases}$$

and typings $x_1 : T_1, \dots, x_k : T_k$ in the current context Γ such that

(1) $x_i^\natural = \psi_i$,

(2) **(Inner)** whenever $[p_j] = [p_i[q]]$ we have $\mathtt{t}\, x_j = \mathtt{s}_{[q]}\, x_i$,

then:

$$\frac{\Gamma}{\Gamma \vdash \begin{cases} [p_1] \leftarrow x_1 \\ \vdots \\ [p_k] \leftarrow x_k \end{cases}} \; \texttt{graft}$$

The shape of this pasting diagram is of course $\mathbf{p}$.

Shift to the next dimension. If we have a pasting diagram $\mathbf{P}$ of shape $\mathbf{P}^\natural = \mathbf{p}$, a cell $x : \mathbf{Q} \longrightarrow a$ in the current context, such that

(1) $x^\natural = \mathtt{t}\, \mathbf{p}$,

(2) **(Glob1)** if $\mathbf{p}$ is non-degenerate, then $\mathtt{t}\mathtt{s}_{[]}\, \mathbf{P} = \mathtt{t}\, x$,

(3) **(Glob2)** if $\mathbf{p}$ is non-degenerate, then for a leaf $[p[q]] \in \mathbf{p}^!$, we have $\mathtt{s}_{[q]}\, \mathtt{s}_{[p]}\, \mathbf{P} = \mathtt{s}_{\wp_{\mathbf{p}}[p[q]]}\, x$,

(4) **(Degen)** if $\mathbf{p}$ is degenerate, then $\mathbf{Q} = \{[] \leftarrow a\,$,

then:

$$\frac{\Gamma \vdash \mathbf{P}}{\Gamma, y : \mathbf{P} \longrightarrow x} \; \texttt{shift}$$

for y a fresh variable.

Remark 4.46. The syntax of OptSet$^?$ is closely related to the notion of FOLDS [20]. In fact, the category $\mathbb{O}^{\mathrm{op}}$ is a FOLDS signature with no relation symbols. Intuitively, each opetope ω corresponds to a sort, that depends on variables a_{f} of type ψ_{f}, where $\mathsf{f} : \omega \longrightarrow \psi_{\mathsf{f}}$ ranges over $\omega/\mathbb{O}^{\mathrm{op}} - \{\mathrm{id}_\omega\} \cong \mathbb{O}/\omega - \{\mathrm{id}_\omega\}$. Since morphisms of $\mathbb{O}^{\mathrm{op}}$ are generated by (opposites of) source and target embeddings, it is enough to consider the type ω to be only parametrized by a_{f} where f is a source or target embedding, modulo additional coherence conditions. For instance, if ω corresponds to the following preopetope

$$\begin{cases} [p_1] \leftarrow \psi_1 \\ \vdots \\ [p_k] \leftarrow \psi_k \end{cases}$$

(the case where ω is degenerate is similar) then the following type in $\textsc{OptSet}^?$

$$\begin{cases} [p_1] \leftarrow x_1 & \\ \vdots & \longrightarrow y \\ [p_k] \leftarrow x_k & \end{cases}$$

precisely represents the sort ω parametrized by variables a_{f} of type ψ_{f}, where $\mathsf{f} : \omega \longrightarrow \psi_{\mathsf{f}}$ is a source or target embedding: $a_{\mathsf{s}_{[p_i]}} := x_i$, and $a_{\mathsf{t}} := y$. The side conditions of rules `graft` and `shift` guarantee that this indeed corresponds to a sort in FOLDS, namely, that we can find dependency variables a_{f} for *all* $\mathsf{f} \in \omega/\mathbb{O}^{\mathrm{op}} - \{\mathrm{id}_\omega\}$.

4.7 Equivalence with opetopic sets

Definition 4.47 (Substitution)**.** Let Υ and Γ be two derivable contexts in $\textsc{OptSet}^?$. Write the context Γ as $(x_1 : T_1, \ldots, x_k : T_k)$. By construction, $1 \leq i \leq k$, the variable x_i does not occur in the type of x_j whenever $j < i$. Akin to classical type theory (see e.g. [14, definition 2.11]), a substitution $\sigma : \Upsilon \longrightarrow \Gamma$ is a sequence of variables $(\sigma_1, \ldots, \sigma_k)$ such that for $1 \leq i \leq k$, the typing $\sigma_i : T_i[\sigma_1/x_1]\cdots[\sigma_{i-1}/x_{i-1}]$ (i.e. T_i where all instances of x_j have been replaced by σ_j, for $1 \leq j < i$) is in Υ.

Write the context Υ as $(y_1 : U_1, \ldots, y_l : U_l)$, and let $\tau = (\tau_1, \ldots, \tau_l) : \Xi \longrightarrow \Upsilon$ be another substitution. For $1 \leq i \leq k$, if $f(i)$ is the unique index such that $y_{f(i)} = \sigma_i$, then the composite $\tau\sigma : \Xi \longrightarrow \Gamma$ is given by the sequence of variables $(\tau_{f(1)}, \ldots, \tau_{f(k)})$.

Let $\mathcal{C}\mathrm{tx}^?$ be the syntactic category of our type theory, i.e. the category whose objects are derivable contexts, and morphisms are substitutions as defined above.

Lemma 4.48. *In the setting above, we have* $\sigma_i^\natural = x_i^\natural$.

Proof. The shape of a variable, i.e. the shape of its source pasting diagram, does not depend on the variables present in it, only on its underlying preopetope. □

Recall from definition 4.26 the *unnamed coding function* $C^? : \mathbb{O}_n \longrightarrow \mathbb{P}_n^\checkmark$.

Definition 4.49 (Unnamed stratification)**.** We construct the *unnamed stratification functor* $S^? : (\mathcal{C}\mathrm{tx}^?)^{\mathrm{op}} \longrightarrow \widehat{\mathbb{O}}_{\mathrm{fin}}$. For $\Gamma \in \mathcal{C}\mathrm{tx}^?$ and $\omega \in \mathbb{O}$, let

$$S^?\Gamma_\omega := \left\{ x \in \Gamma \mid x^\natural = C^?(\omega) \right\}.$$

If $x^\natural \neq \blacklozenge$, then the type X of x is of the form $\mathbf{P} \longrightarrow z$, and we let $\mathsf{t}\, x := z$. This is well defined as by construction of Γ we have $z^\natural = \mathsf{t}(x^\natural)$. For $[p] \in \omega^\bullet$, we let $\mathsf{s}_{[p]}\, x := \mathsf{s}_{[p]}\, \mathbf{P}$. Again, this is well-defined as $(\mathsf{s}_{[p]}\, \mathbf{P})^\natural = \mathsf{s}_{[p]}(\mathbf{P}^\natural) = \mathsf{s}_{[p]}(x^\natural)$. From there, the opetopic identities clearly hold, and $S^?\Gamma$ is a finite opetopic set.

On morphisms, write $\Gamma = (x_1 : T_1, \ldots, x_k : T_k)$, let $\sigma = (\sigma_1, \ldots, \sigma_k) : \Upsilon \longrightarrow \Gamma$ be a substitution, and define a morphism $S^?\sigma : S^?\Gamma \longrightarrow S^?\Upsilon$ as follows. For x_i a variable of Γ, and $\omega \in \mathbb{O}$ such that $C^?(\omega) = x_i^\natural$, there is a corresponding cell $x_i \in S^?\Gamma_\omega$, and we let $(S^?\sigma)(x_i) := \sigma_i$. This is well-defined since by lemma lemma 4.48, we have $\sigma_i^\natural = x^\natural = \omega$, thus $\sigma_i \in S^?\Upsilon_\omega$.

Lemma 4.50. *The map* $S^?\sigma$ *of definition 4.49 is a morphism of opetopic sets.*

Proof. Assume $\omega \neq \blacklozenge$, so that the type of x_i is $\mathbf{P} \longrightarrow x_j$ for some $j < i$, and the type of σ_i is $\mathbf{P}[\sigma_1/x_1]\cdots[\sigma_{i-1}/x_{i-1}] \longrightarrow \sigma_j$. Then $(S^?\sigma)(\mathtt{t}\, x_i) = \sigma_j = \mathtt{t}(S^?\sigma)(x_i)$. If $[p] \in \omega^\bullet$, then $\mathtt{s}_{[p]}\, x_i = x_l$, for some $l < i$, and

$$\begin{aligned}(S^?\sigma)(\mathtt{s}_{[p]}\, x_i) &= (S^?\sigma)(x_l)\\ &= \sigma_l && \text{see definition 4.47}\\ &= \mathtt{s}_{[p]}\,(\mathbf{P}[\sigma_1/x_1]\cdots[\sigma_l/x_l]\cdots[\sigma_{i-1}/x_{i-1}])\\ &= \mathtt{s}_{[p]}(S^?\sigma(x_i)) && \text{see definition 4.47.}\end{aligned}$$

In conclusion, $S^?\sigma$ is compatible with the source and target maps, and thus is a morphism of opetopic sets $S^?\Gamma \longrightarrow S^?\Upsilon$. □

Theorem 4.51. *The stratification functor $S^? : (\mathcal{C}\mathrm{tx}^?)^{\mathrm{op}} \longrightarrow \widehat{\mathbb{O}}_{\mathrm{fin}}$ is an equivalence of categories.*

Proof. It is clear from the definition that $S^?$ is a faithful functor. Let $\Gamma, \Upsilon \in \mathcal{C}\mathrm{tx}^?$ be derivable contexts, write Γ as $(x_1 : T_1, \ldots, x_k : T_k)$, and $f : S^?\Gamma \longrightarrow S^?\Upsilon$. For σ_f the substitution $(f(x_1), \ldots, f(x_k)) : \Gamma \longrightarrow \Upsilon$, we clearly have $f = S^?\sigma_f$, showing that $S^?$ is fully faithful.

We now show that $S^?$ is essentially surjective. Take $X \in \widehat{\mathbb{O}}_{\mathrm{fin}}$, and enumerate its cells as $x_1 \in X_{\omega_1}, \ldots, x_k \in X_{\omega_k}$, such that whenever $i < j$ we have $\dim \omega_i \leq \dim \omega_j$. In other words, they are sorted by dimension. We produce a sequence of derivable contexts $\Gamma^{(0)} \subseteq \Gamma^{(1)} \subseteq \cdots \subseteq \Gamma^{(k)}$, where $\Gamma^{(i)} = (\overline{x_1} : T_1, \ldots, \overline{x_i} : T_i)$ is such that $\overline{x_i}^\natural = C^?(\omega_i) = C^?\left(x_i^\natural\right)$. For $i = 0$, let $\Gamma^{(0)} = ()$. Assume $1 \leq i \leq k$, and that $\Gamma^{(i-1)}$ is defined and derivable.

(1) If $\omega_i = \blacklozenge$, let $\Gamma^{(i)}$ be given by the following proof tree:

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma^{(i-1)}\end{array}}{\Gamma^{(i-1)}, \overline{x_i} : \blacklozenge}\ \mathtt{point}$$

Note that in this case, $T_i = \blacklozenge$.

(2) Assume $\omega_i \neq \blacklozenge$ is not degenerate. By induction $\overline{x_j}^\natural = C^?\left(x_j^\natural\right)$ for all $1 \leq j < i$. In particular, for every address $[p] \in \omega_i^\bullet$, we have $\overline{\mathtt{s}_{[p]}\, x_i}^\natural = \mathtt{s}_{[p]}\, C^?(\omega_i)$. Further, $\overline{\mathtt{t}\, x_i}^\natural = C^?\left((\mathtt{t}\, x_i)^\natural\right) = \mathtt{t}\, C^?\left(x_i^\natural\right)$. We define $\Gamma^{(i)}$ by the following proof tree:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma^{(i-1)}\end{array}}{\Gamma^{(i-1)} \vdash \begin{cases}[p_1] \leftarrow \overline{\mathtt{s}_{[p_1]}\, x_i}\\ \vdots\end{cases}}\ \mathtt{graft}}{\Gamma^{(i-1)}, \overline{x_i} : \begin{cases}[p_1] \leftarrow \overline{\mathtt{s}_{[p_1]}\, x_i}\\ \vdots\end{cases} \longrightarrow \overline{\mathtt{t}\, x_i}}\ \mathtt{shift}$$

where the pasting diagram has shape $C^?(\omega)$, and $\{[p_1], \ldots\} := \omega_i^\bullet$. The side conditions of `graft` and `shift` are met since X is an opetopic set.

(3) If ω_i is degenerate, then $\Gamma^{(i)}$ is given by the following proof tree:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma^{(i-1)}\end{array}}{\Gamma^{(i-1)} \vdash \{\!\{\overline{\mathtt{t}\,\mathtt{t}\, x_i}}\ \mathtt{degen}}{\Gamma^{(i-1)}, \overline{x_i} : \{\!\{\overline{\mathtt{t}\,\mathtt{t}\, x_i} \longrightarrow \overline{\mathtt{t}\, x_i}}\ \mathtt{shift}$$

Finally, the mapping $x_i \longmapsto \overline{x_i}$ exhibits an isomorphism $X \longrightarrow S^?\Gamma^{(k)}$, and $S^?$ is essentially surjective. □

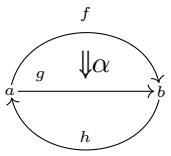
The category $\mathcal{C}\mathrm{tx}^?$ has finite limits, induced from finite colimits in $\widehat{\mathbb{O}}_{\mathrm{fin}}$ through $S^?$. We conclude this section with a result similar to theorem 3.102, stating that system OptSet$^!$ is a theory for opetopic sets.

Theorem 4.52. *We have an equivalence* $\widehat{\mathbb{O}} \simeq \mathcal{L}\mathrm{ex}(\mathcal{C}\mathrm{tx}^?, \mathcal{S}\mathrm{et})$.

Proof. This follows directly from theorem 4.51 and from the Gabriel–Ulmer duality [6] [17, theorem 8]. □

4.8 Examples In this section, we give example derivations in system OptSet$^?$. For clarity, we do not repeat the type of previously typed variables in proof trees.

Example 4.53. We show how to derive the following opetopic set, which is not representable:



First, we introduce all the points:

$$\dfrac{\dfrac{}{a : \blacklozenge}\,\texttt{point}}{a, b : \blacklozenge}\,\texttt{point}$$

Then we introduce f, by first specifying its source pasting diagram with the `graft` rule, parameterized by $\blacksquare = \{* \leftarrow \blacklozenge$, and then applying the `shift` rule:

$$\dfrac{\dfrac{\vdots}{a, b}\,\texttt{graft}}{a, b \vdash \{* \leftarrow a}\,\texttt{shift}}{a, b, f : \{* \leftarrow a \longrightarrow b}$$

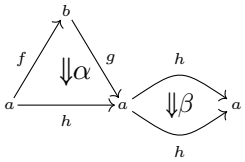
We proceed similarly for g and h:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{a, b, f}\,\texttt{graft}}{a, b, f \vdash \{* \leftarrow a}\,\texttt{shift}}{a, b, f, g : \{* \leftarrow a \longrightarrow b}\,\texttt{graft}}{a, b, f, g \vdash \{* \leftarrow b}\,\texttt{shift}}{a, b, f, g, h : \{* \leftarrow b \longrightarrow a}$$

Lastly, we introduce α, first by specifying its source with the `graft` rule, parameterized by $\mathbf{1} = \{[] \leftarrow \blacksquare$ (see the opetopic integers defined in example 4.36), and applying the `shift` rule:

$$\dfrac{\dfrac{\dfrac{\vdots}{a, b, f, g, h}\,\texttt{graft}}{a, b, f, g, h \vdash \{[] \leftarrow f}\,\texttt{shift}}{a, b, f, g, h, \alpha : \{[] \leftarrow f \longrightarrow g}$$

Example 4.54. The opetopic set



is straightforwardly derived as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a:\blacklozenge}\,\texttt{point}}{a,b:\blacklozenge}\,\texttt{point}}{a,b\vdash\{*\leftarrow a}\,\texttt{graft}}{a,b,f:\{*\leftarrow a\longrightarrow b}\,\texttt{shift}}{a,b,f\vdash\{*\leftarrow b}\,\texttt{graft}}{a,b,f,g:\{*\leftarrow b\longrightarrow a}\,\texttt{shift}}{a,b,f,g\vdash\{*\leftarrow a}\,\texttt{graft}}{a,b,f,g,h:\{*\leftarrow a\longrightarrow a}\,\texttt{shift}}{a,b,f,g,h\vdash\begin{cases}[]\leftarrow g\\[*]\leftarrow f\end{cases}}\,\texttt{graft}}{a,b,f,g,h,\alpha:\begin{cases}[]\leftarrow g\\[*]\leftarrow f\end{cases}\longrightarrow h}\,\texttt{shift}}{a,b,f,g,h,\alpha\vdash\{[]\leftarrow h}\,\texttt{graft}}{a,b,f,g,h,\alpha,\beta:\{[]\leftarrow h\longrightarrow h}\,\texttt{shift}$$

Example 4.55 (A classic, maximally folded)**.** We derive the following opetopic set

where all 0-cells are the same cell a, all 1-cells are f, the 2-cells on the left are α, the 2-cell on the right is β, and the 3-cell is A. Note that those identifications make the opetopic set not representable. We first derive a and f:

$$\dfrac{\dfrac{\dfrac{}{a:\blacklozenge\ \ \mathsf{context}}\,\texttt{point}}{a\vdash\{*\leftarrow a}\,\texttt{graft}}{a,f:\{*\leftarrow a\longrightarrow a\ \ \mathsf{context}}\,\texttt{shift}$$

Then we introduce α, by first specifying its source pasting diagram with the `graft` rule, parameterized by opetope $\mathbf{2} = \begin{cases}[]\leftarrow\blacksquare\\[*]\leftarrow\blacksquare\end{cases}$ (see the opetopic integers defined in example 4.36), and applying the `shift` rule:

$$\dfrac{\dfrac{\vdots}{a,f\ \ \mathsf{context}}\,\texttt{graft}}{a,f\vdash\begin{cases}[]\leftarrow f\\[*]\leftarrow f\end{cases}}\,\texttt{shift}$$
$$a,f,\alpha:\begin{cases}[]\leftarrow f\\[*]\leftarrow f\end{cases}\longrightarrow f\ \ \mathsf{context}$$

Likewise, we introduce β, where the `graft` rule is parameterized by **3**:

$$\dfrac{\dfrac{\vdots}{a, f, \alpha \ \ \mathsf{context}}}{a, f, \alpha \vdash \begin{cases} [] \leftarrow f \\ [*] \leftarrow f \\ [**] \leftarrow f \end{cases}}\ \texttt{graft}$$

$$\dfrac{\qquad\qquad}{a, f, \alpha, \beta : \begin{cases} [] \leftarrow f \\ [*] \leftarrow f \quad \longrightarrow f \ \ \mathsf{context} \\ [**] \leftarrow f \end{cases}}\ \texttt{shift}$$

Lastly, we introduce A, where the `graft` rule is parameterized by $\begin{cases} [] \leftarrow \mathbf{2} \\ [[*]] \leftarrow \mathbf{2} \end{cases}$:

$$\dfrac{\dfrac{\vdots}{a, f, \alpha, \beta \ \ \mathsf{context}}}{a, f, \alpha, \beta \vdash \begin{cases} [] \leftarrow \alpha \\ [[*]] \leftarrow \alpha \end{cases}}\ \texttt{graft}$$

$$\dfrac{\qquad\qquad}{a, f, \alpha, \beta, A : \begin{cases} [] \leftarrow \alpha \\ [[*]] \leftarrow \alpha \end{cases} \longrightarrow \beta \ \ \mathsf{context}}\ \texttt{shift}$$

4.9 Python implementation System OptSet$^?$ is implemented in the Python module called `opetopy.UnnamedOpetopicSet` of [13]. The rules are represented by functions `point`, `degen`, `graft`, and `shift`, and are further encapsulated in rule instance classes `Point`, `Degen`, `Graft`, and `shift`.

Figure 4.5: Derivation of example 4.55 using `opetopy.NamedOpetope`

```
from opetopy.UnnamedOpetopicSet import *
from opetopy.UnnamedOpetope import address, Arrow, OpetopicInteger
from opetopy.UnnamedOpetope import Graft as OptGraft
from opetopy.UnnamedOpetope import Shift as OptShift
# We first derive the unique point a.
classic = Point("a")
# We then derive f by firstly specifying a pasting diagram with the graft rule. It
↪  is constructed with a proof tree of its shape opetope (in system Opt?), and an
↪  address-to-variable mapping.
classic = Graft(
    pastingDiagram(
        Arrow(),
        {address([], 0): "a"}), # ∗ ← a
    classic)
# We then derive f.
classic = Fill("a", "f", classic)
# In a similar way, we derive α of shape 2.
classic = Graft(
    pastingDiagram(
        OpetopicInteger(2),
        {
            address([], 1): "f", # [] ← f
            address(["*"]): "f" # [∗] ← f
        }),
    classic)
classic = Fill("f", "alpha", classic)
# In a similar way, we derive β of shape 3.
classic = Graft(
    pastingDiagram(
        OpetopicInteger(3),
        {
            address([], 1): "f", # [] ← f
            address(["*"]): "f", # [∗] ← f
            address(["*", "*"]): "f" # [∗∗] ← f
        }),
    classic)
classic = Fill("f", "beta", classic)
# We now take a break to derive ω = Y2 ∘[[∗]] Y2, the shape of A, in system Opt?.
omega = OptGraft(
    OptShift(OpetopicInteger(2)),
    OpetopicInteger(2),
    address([["*"]]))
# Finally, we derive A of shape ω.
classic = Graft(
    pastingDiagram(
        omega,
        {
            address([], 2): "alpha", # [] ← α
            address([["*"]]): "alpha" # [[∗]] ← f
        }),
    classic)
classic = Fill("beta", "A", classic)
```

Appendix A: Polynomial monads and the Baez–Dolan $(-)^+$ construction

Starting from a polynomial monad M with set of operations B, the Baez–Dolan construction gives a new polynomial monad M^+ having B as its set of colors. In this chapter, we study the monad structure of M^+ in depth.

A.1 Polynomial monads In this section, we show that three definitions of polynomial monads are equivalent: the "official one" as a monoid object in $\mathcal{P}\mathrm{oly}\mathcal{E}\mathrm{nd}(I)$ (see definition 2.23), the definition as $(-)^\star$-algebra (see definition 2.24), and the definition by so-called partial laws (in the style of partial composition in operads or placed composition in multicategories).

Definition A.1 (Partial laws)**.** Let P be the following polynomial endofunctor:

$$I \xleftarrow{\ s\ } E \xrightarrow{\ p\ } B \xrightarrow{\ t\ } I.$$

A set of *partial laws* on P is the datum of the following:

(Unit) a map $\eta : I \longrightarrow B$;

(Partial multiplication) a map $\wedge : E \times_I B \longrightarrow B$, where for $(e, b) \in E \times_I B$ and $a := p(e)$ we write $a \wedge_e b$ instead of $\wedge(e, b)$, and say that the expression $a \wedge_e b$ is *admissible*; this will be tacitly assumed from now on;

(Partial readdressing) for a, b, and e as above, a bijective map $\rho_{a \wedge_e b} : (E(a) - \{e\}) + E(b) \longrightarrow E(a \wedge_e b)$ over I;

such that the following conditions are satisfied:

(Trivial) for $i \in I$, we have $t(\eta(i)) = i$, and $E(\eta(i))$ is a singleton whose unique element e is such that $s(e) = i$;

(Left unit) for $i \in I$, $b \in B$, and e the unique element of $E(\eta(i))$, we have[8] $\eta(i) \wedge_e b = b$, and $\rho_{\eta(i) \wedge_e b} : E(b) \longrightarrow E(b)$ is the identity;

(Right unit) for $i \in I$, $b \in B$, and $e \in E(b)$, we have $b \wedge_e \eta(i) = b$, and $\rho_{b \wedge_e \eta(i)}$ is given by

$$\begin{aligned} \rho_{b \wedge_e \eta(i)} : (E(b) - \{e\}) + E(\eta(i)) &\longrightarrow E(b) \\ x \in E(b) - \{e\} &\longmapsto x \\ y \in E(\eta(i)) &\longmapsto e; \end{aligned}$$

(Disjoint multiplication) for $a \wedge_e b$ and $a \wedge_f c$, where $e \neq f$, we have

$$(a \underset{e}{\wedge} b) \underset{f'}{\wedge} c \;=\; (a \underset{f}{\wedge} c) \underset{e'}{\wedge} b,$$

where $f' := \rho_{a \wedge_e b}(f)$ and $e' := \rho_{a \wedge_f c}(e)$, and the following coherence diagram commutes:

$$\begin{array}{ccc} (E(a) - \{e, f\}) + E(b) + E(c) & \xrightarrow{\rho_{a \wedge_e b}} & (E(a \wedge_e b) - \{f'\}) + E(c) \\ \downarrow{\scriptstyle \rho_{a \wedge_f c}} & & \downarrow{\scriptstyle \rho_{(a \wedge_e b) \wedge_{f'} c}} \\ (E(a \wedge_f c) - \{e'\}) + E(b) & \xrightarrow{\rho_{(a \wedge_e b) \wedge_{f'} c}} & E((a \wedge_e b) \wedge_{f'} c); \end{array} \tag{A.2}$$

(Nested multiplication) for $a \wedge_e b$ and $b \wedge_f c$, we have

$$(a \underset{e}{\wedge} b) \underset{f'}{\wedge} c \;=\; a \underset{e}{\wedge} (b \underset{f}{\wedge} c),$$

[8] Recall that expressions of the form $\rho_{a \wedge_e b}$ are tacitly admissible. In this case, it means that $t(b) = s(e) = i$.

where $f' := \rho_{a \wedge_e b}(f)$, and the following coherence diagram commutes:

$$\begin{array}{ccc} (E(a) - \{e\}) + (E(b) - \{f\}) + E(c) & \xrightarrow{\rho_{a \wedge_e b}} & (E(a \wedge_e b) - \{f'\}) + E(c) \\ \downarrow{\scriptstyle \rho_{b \wedge_f c}} & & \downarrow{\scriptstyle \rho_{a \wedge_e (b \wedge_f c)}} \\ (E(a) - \{e\}) + E(b \wedge_f c) & \xrightarrow{\rho_{(a \wedge_e b) \wedge_{f'} c}} & E((a \wedge_e b) \wedge_{f'} c). \end{array} \tag{A.3}$$

Proposition A.4. *Let P be a polynomial functor with partial laws as in definition A.1. Then P is naturally a polynomial monad. Specifically,*

(1) the unit morphism $\eta : \mathrm{id}_{\mathbf{Set}/I} \longrightarrow P$ is given by **(Unit)***;*

(2) the law $\mu : PP \longrightarrow P$ is given by repeated applications of **(Partial multiplication)***; in details, recall from remark 2.22 that the operations of PP are the P-trees of uniform height 2, i.e. of the form*

$$T = \mathsf{Y}_a \bigcirc_i \mathsf{Y}_{b_i},$$

where $E(a) = \{e_1, \dots, e_k\}$; from here

$$\mu_1(T) := (\ \cdots\ (a \underset{e_1}{\wedge} b_1) \underset{e_2}{\wedge} b_2\ \cdots\ ) \underset{e_k}{\wedge} b_k.$$

Proof. We must check that the following classical diagrams commute.

$$\begin{array}{ccc} P & \xrightarrow{\eta P} & PP \\ & \searrow & \downarrow{\scriptstyle \mu} \\ & & P, \end{array} \qquad \begin{array}{ccc} P & \xrightarrow{P\eta} & PP \\ & \searrow & \downarrow{\scriptstyle \mu} \\ & & P, \end{array} \qquad \begin{array}{ccc} PPP & \xrightarrow{\mu P} & PP \\ {\scriptstyle P\mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ PP & \xrightarrow{\mu} & P. \end{array} \tag{A.5}$$

The main arguments go as follows.

(1) In the first diagram, the morphism ηP maps an operation $b \in B$ to $\mathsf{I}_i \circ_{[]} \mathsf{Y}_b = \eta_1(i) \wedge_e b$, where e is the unique element of $E(i)$. Thus, the triangle commutes by **(Left unit)**.

(2) In the second diagram, the morphism $P\eta$ maps an operation $b \in B$ to

$$\mathsf{Y}_b \bigcirc_{e \in E(b)} \mathsf{I}_{s(e)} = \Big(\cdots \Big(b \underset{e_1}{\wedge} \eta_1(s(e_1)) \Big) \underset{e_2}{\wedge} \eta_1(s(e_2)) \cdots \Big) \underset{e_k}{\wedge} \eta_1(s(e_k)),$$

where $E(b) = \{e_1, \dots, e_k\}$. By **(Right unit)**, $b \wedge_{e_1} \eta_1(s(e_1)) = b$. By iterated application of this rule, we successively eliminate the $\eta_1(s(e_i))$'s and the whole expression reduces to b. In addition, the final readdressing map $\mathsf{Y}_b \longrightarrow \mathsf{Y}_b$ is the identity. In conclusion the second triangle commutes.

(3) By iterating the construction of remark 2.22, the operations of PPP are P-trees of uniform height 3. Writing down such a tree is tedious, but much like the previous two cases, commutativity of the third diagram of equation (A.5) follows from repeated applications of **(Disjoint multiplication)** and **(Nested multiplication)**. The coherence diagrams (A.2) and (A.3) ensure that the readdressing map μ_2 does not depend on the order in which these rules are applied. □

Proposition A.6. *A polynomial monad (P, η, μ) is canonically a $(-)^\star$-algebra.*

Proof. Let $T \in \mathrm{tr}\, P$. Recall that a branch of T is a path from the root edge to a leaf of T, or equivalently, a leaf address of T. The length of a branch $[l] = [e_1\, e_2 \cdots e_N] \in T^!$ is the number of elements of the sequence l, in this case N. Recall that a P-tree T is *uniform* of all its branches have the same length. We now define the *uniformization* operation $\overline{(-)} : \mathrm{tr}\, P \longrightarrow \mathrm{tr}\, P$, together with, for each $T \in \mathrm{tr}\, P$, a bijection $c_T : T^! \longrightarrow \bar{T}^!$ over I, inductively as follows.

(1) Trivially, if T is already uniform (e.g. I_i with $i \in I$, Y_b with $b \in B$), then we set $\bar{T} := T$ and c_T to be the identity.

(2) Assume that T is not uniform, and let N be the length of a longest branch. Let $[l_1], \ldots, [l_k] \in T^!$ be the non maximal branches of T, of length $s_1, \ldots, s_k < N$ respectively. Further, let $i_j := \mathsf{e}_{[l_i]} T \in I$ be the decoration of the leaf at address $[l_i]$. Recall that the corolla $\mathsf{Y}_{\eta(i_j)}$ is unary, i.e. $\mathsf{Y}^!_{\eta(i_j)} = \{[[]]\}$. Define

$$\bar{T} = T \bigcirc_{[l_i]} \underbrace{\mathsf{Y}_{\eta(i_j)} \underset{[[]]}{\circ} \mathsf{Y}_{\eta(i_j)} \underset{[[]]}{\circ} \cdots \underset{[[]]}{\circ} \mathsf{Y}_{\eta(i_j)}}_{N-k_i \text{ times}}$$

and

$$\begin{aligned} c_T : T^! &\longrightarrow \bar{T}^! \\ [l] \text{ max. branch} &\longmapsto [l] \\ [l_i] \text{ non-max. branch} &\longmapsto [l_i \underbrace{[]\,[]\cdots[]}_{N-k_i}] \end{aligned}$$

In short, $\overline{(-)} : \mathrm{tr}\, P \longrightarrow \mathrm{tr}\, P$ "uniformizes" T into a tree $\bar{T}$ of uniform height N by grafting corollas of the form $\mathsf{Y}_{\eta(i)}$ onto non-maximal branches. Since these corollas are unary the correspondence c_T is quite clear. Now, the structure map

$$\begin{array}{ccccccc} I & \longleftarrow & \mathrm{tr}^!\, P & \longrightarrow & \mathrm{tr}\, P & \longrightarrow & I \\ \| & & \wp \downarrow & \lrcorner & \downarrow \mathsf{t} & & \| \\ I & \xleftarrow{s} & E & \xrightarrow{p} & B & \xrightarrow{t} & I \end{array}$$

can be defined. For $T \in \mathrm{tr}\, P$, since $\bar{T}$ has uniform height, say N, it is an operation of $P^N = PP\cdots P$, and $\mathsf{t}P$ is obtained by repeated application of μ_1, and the readdressing map $\wp_T$ is given by c_T followed by repeated applications of μ_2. From here, it is a tedious but straightforward process to show that the $(-)^\star$-algebra conditions (i.e. the diagrams in the proof of lemma A.7) follow from the monad conditions on (P, η, μ). □

Lemma A.7 (Contraction associativity formula)**.** *Let P be a $(-)^\star$-algebra as in*

$$\begin{array}{ccccccc} I & \longleftarrow & \mathrm{tr}^!\, P & \longrightarrow & \mathrm{tr}\, P & \longrightarrow & I \\ \| & & \wp \downarrow & \lrcorner & \downarrow \mathsf{t} & & \| \\ I & \xleftarrow{s} & E & \xrightarrow{p} & B & \xrightarrow{t} & I \end{array} \tag{A.8}$$

Let t and $\wp$ be as in definition 2.27.

(1) If $b \in B$, then $\mathsf{t}\mathsf{Y}_b = b$. Recall from remark 2.12 that the set of leaf address of Y_b is simply $\{[e] \mid e \in E(b)\}$, and $\wp_{\mathsf{Y}_b}$ maps $[e]$ to e.

(2) If we have a grafting $T \circ_{[l]} U$ of P-trees, then

$$\mathsf{t}(T \underset{[l]}{\circ} U) \;=\; \mathsf{t}(\mathsf{Y}_{\mathsf{t}T} \underset{[\wp_T[l]]}{\circ} \mathsf{Y}_{\mathsf{t}U}). \tag{A.9}$$

Further, for $[r] \in U^!$, we have a leaf $[lr] \in \left(T \circ_{[l]} U\right)^!$, and writing $V := \mathsf{Y}_{\mathsf{t}T} \circ_{[\wp_T[l]]} \mathsf{Y}_{\mathsf{t}U}$, we have

$$\wp_{T \circ_{[l]} U}[lr] \;=\; \wp_V\left(\wp_T[l] \cdot \wp_U[r]\right). \tag{A.10}$$

Proof. Since P is an algebra over the monad $(-)^\star$, the following two diagrams commute:

$$\begin{array}{ccc} P & \xrightarrow{\eta} & P^\star \\ & \searrow & \downarrow{\scriptstyle m} \\ & & P, \end{array} \qquad \begin{array}{ccc} P^{\star\star} & \xrightarrow{m^\star} & P^\star \\ {\scriptstyle \mu}\downarrow & & \downarrow{\scriptstyle m} \\ P^\star & \xrightarrow{m} & P, \end{array}$$

where $m : P^\star \longrightarrow P$ is the structure map of P. The result follows by (tediously) unfolding these diagrams, together with the definition of η and μ given in definition 2.24. □

Proposition A.11. *A $(-)^\star$-algebra, $m : P^\star \longrightarrow P$ has partial laws.*

Proof. Let the function η of **(Unit)** map $i \in I$ to $m_1(\mathsf{l}_i) \in B$. For $(e, b) \in E \times_I B$ and $a := p(e)$, let $T := \mathsf{Y}_a \circ_{[e]} \mathsf{Y}_b$, and $a \wedge_e b := m_1(T)$. Note that

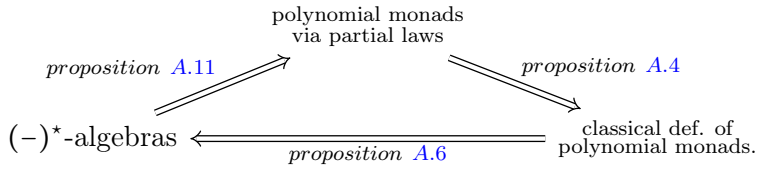
$$T^! = \{[f] \mid f \in E(a), f \neq e\} \cup \{[ef] \mid f \in E(b)\}.$$

From here, one can check that the partial readdressing is simply given by

$$\begin{aligned} \rho_{a \wedge_e b} : (E(a) - \{e\}) + E(b) &\longrightarrow E(a \underset{e}{\wedge} b) \\ [f], f \in E(a) - \{e\} &\longmapsto \wp_T[f] \\ [ef], f \in E(b) &\longmapsto \wp_T[ef], \end{aligned}$$

and that the required properties follow from lemma A.7. □

Remark A.12. Combining the results of propositions A.4, A.6 and A.11, we obtain the following triangle of implications, featuring from left to right, *total composition* (the composition of a whole tree in one shot), *partial composition* (or placed composition), and *classical composition* (the composition of a tree of uniform height 2):



It can be shown that each implication is the object part of a functor, and that these functors are isomorphisms (the inverse of each being the composite of the other two). The situation is similar to the three definitions of symmetric operad given in [19, sections 5.3 and 5.5], namely *combinatorial*, *partial* and *classical*. As a matter of fact, polynomial monads are the same as coloured operads with a free action of the symmetric group.

A.2 A complete $(-)^+$ construction Recall from definition 2.29 that if M is a polynomial monad

$$I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} I,$$

then the underlying polynomial endofunctor of M^+ is

$$B \xleftarrow{\mathsf{s}} \mathrm{tr}^\bullet M \xrightarrow{u} \mathrm{tr}\, M \xrightarrow{\mathsf{t}} B, \tag{A.13}$$

where for $T \in \mathrm{tr}\, M$, the fiber $u^{-1}T$ (which we shall also denote by $T^\bullet$) is the set of node addresses in T, and for $[p] \in T^\bullet$, $\mathsf{s}[p] := \mathsf{s}_{[p]}\, T$.

Theorem A.14 ([16, section 3.2])**.** *The polynomial endofunctor M^+ has a canonical structure of a polynomial monad. Using the definition by partial laws (definition A.1):*

(Unit) *the unit $B \longrightarrow \operatorname{tr} M$ maps b to Y_b;*

(Partial multiplication) *the partial multiplication $\wedge : \operatorname{tr}^\bullet M \times_B \operatorname{tr} M \longrightarrow \operatorname{tr} M$ is given by substitution of trees (see definition 2.19), i.e. for $U \wedge_{[p]} T$ an admissible expression,*

$$U \underset{[p]}{\wedge} T := U \underset{[p]}{\square} T;$$

(Partial readdressing) *for $U \square_{[p]} T$ admissible, define $\rho_{U \square_{[p]} T}$ by (see remark A.18 for a graphical explanation)*

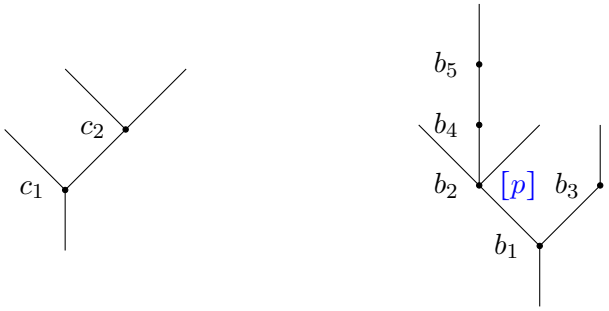
$$(U^\bullet - \{[p]\}) + T^\bullet \longrightarrow (U \underset{[p]}{\square} T)^\bullet$$

$$[q] \in T^\bullet \longmapsto [pq] \tag{A.15}$$

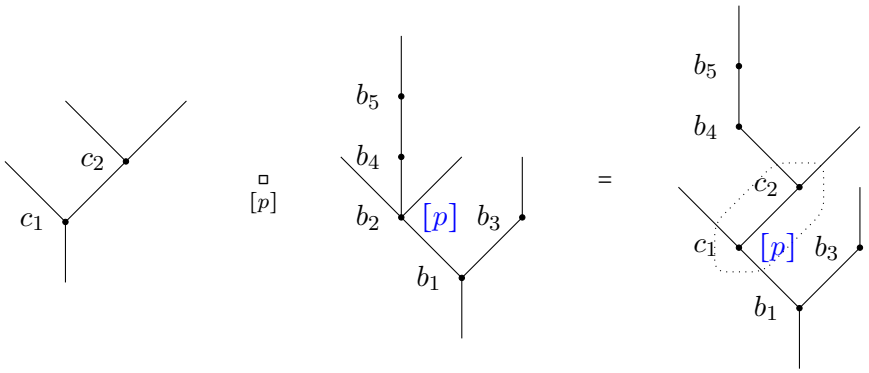
$$[p[e]p'] \in U^\bullet \longmapsto [p] \cdot \wp_T^{-1}(e) \cdot [p'] \tag{A.16}$$

$$[q] \in U^\bullet, [p] \notin [q] \longmapsto [q]. \tag{A.17}$$

Remark A.18. Let T and U be M^+-trees as below respectively (we omit the decorations for simplicity), and $[p] \in U^\bullet$ the address of a node of U, say b_2, written in blue on the right:



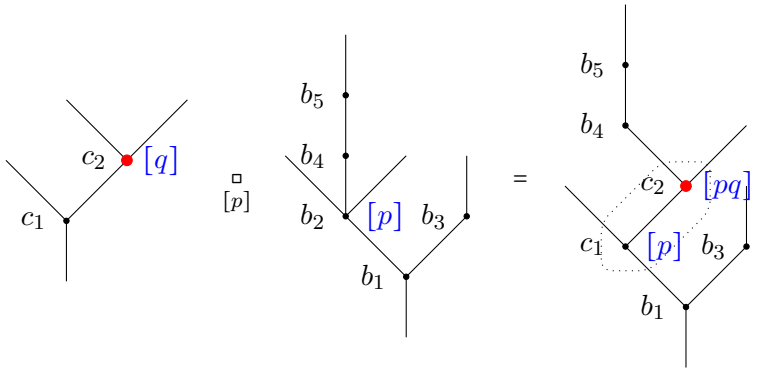
Assuming $\mathsf{t}T = \mathsf{s}_{[p]} U$, the expression $U \square_{[p]} T$ is admissible, and its evaluation gives



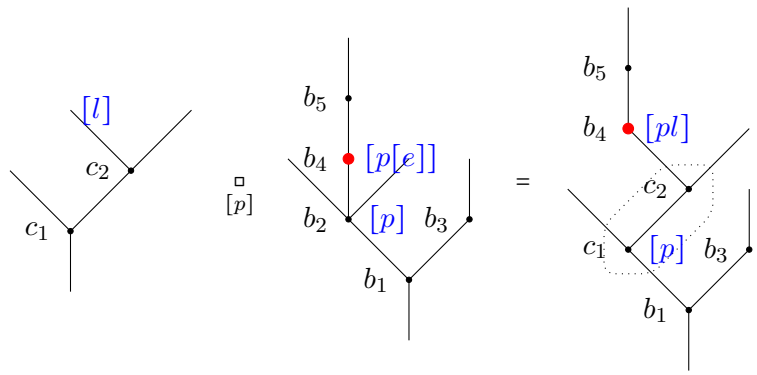
The map $\rho_{U \square_{[p]} T}$ establishes a bijection between $U^\bullet + T^\bullet - \{[p]\}$ and the set of node addresses of $U \square_{[p]} T$. Its definition is based on three cases.

Case (A.15). If $[q] \in T^\bullet$, then the address of the corresponding node in $U \square_{[p]} T$ is simply

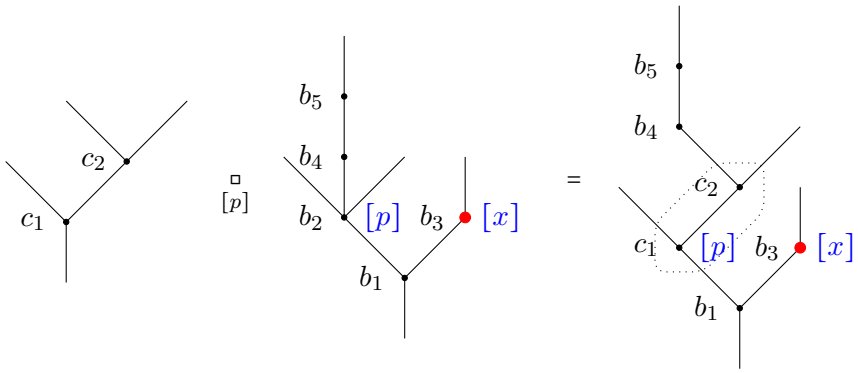
$[pq]$. This reflects that the tree T has been inserted in U at address $[p]$.



Case (A.16). If a node d of U is located "above" the node that will be replaced by T, i.e. if $[p] \sqsubsetneq \&\, d$, then $\&\, d$ necessarily decomposes as $\&\, d = [p[e]p']$, where $[e] \in (\mathsf{s}_{[p]}\, U)^\bullet$. In the example below, $d = b_4$, so that $[p'] = []$. On the other hand, there is a bijection $\wp_T$ between the leaves of T and the input edges of b_2. Assuming $\wp_T^{-1}[e] = [l]$, the new address of b_4 in $U \,\square_{[p]}\, T$ is $[p] \cdot \wp_T^{-1}(e) \cdot [p'] = [plp']$.



Case (A.17). The last case concerns nodes of U that are not located above b_2, and states that their address does not change.



Proof of theorem A.14, **(Trivial)**. For $b \in B$ we have $\mathsf{t}\, \mathsf{Y}_b = b$. On the other hand, $\mathsf{Y}_b^\bullet = \{[]\}$ and $\mathsf{s}_{[]}\, \mathsf{Y}_b = b$. □

Proof of **(Left unit)**. Let $b \in B$ and $T \in \mathrm{tr}\, M$ be such that $\mathsf{Y}_b \,\square_{[]}\, T$ is admissible. Then, by definition, $\mathsf{Y}_b \,\square_{[]}\, T = T$. Then, $\mathsf{Y}_b^\bullet + T^\bullet - \{[]\} = T^\bullet$, and $\rho_{\mathsf{Y}_b \,\square_{[]}\, T}$ maps $[q] \in T^\bullet$ to $[q]$, so it is indeed the identity. □

Proof of **(Right unit)**. Let $b \in B$, $T \in \operatorname{tr} M$, and $[p] \in T^\bullet$ be such that $T \,\square_{[p]}\, \mathsf{Y}_b$ is admissible. By definition, $T \,\square_{[p]}\, \mathsf{Y}_b = T$. Then, $\rho_{T \,\square_{[p]}\, \mathsf{Y}_b}$ is given by

$$\begin{aligned}
(T^\bullet - \{[p]\}) + \{[]\} &\longrightarrow T^\bullet \\
[pep'] \in T^\bullet - \{[p]\} &\longmapsto [p] \cdot \rho_{\mathsf{Y}_b}^{-1}\, e \cdot [p'] \;=\; [pep'] \\
[p'] \in T^\bullet - \{[p]\} \text{ not as above} &\longmapsto [p'] \\
[] \in \mathsf{Y}_b^\bullet &\longmapsto [p]
\end{aligned}$$

as indeed ρ_{Y_b} maps $[e] \in (\mathsf{Y}_b)^{\mathsf{l}}$ to $e \in E(b)$. □

Proof of **(Disjoint multiplication)**. Let $A \,\square_{[e]}\, B$ and $A \,\square_{[f]}\, C$ be two admissible expressions, where $[e] \neq [f]$. Without loss of generality, we distinguish two cases: one where $[e] \sqsubseteq [f]$, and one where $[e]$ and $[f]$ are $\sqsubseteq$-incomparable.

(1) Assume $[e] \sqsubseteq [f]$, so that $[f] = [eqr]$ for some e and r, and decompose A as

$$A \;=\; X \underset{[e]}{\circ} (\mathsf{Y}_{\mathtt{t}B} \bigcirc_{[v_i]} V_i) \underset{[q]}{\circ} Y \underset{[r]}{\circ} \mathsf{Y}_{\mathtt{t}C} \bigcirc_{[w_j]} Z_j,$$

where $q \in E(\mathtt{t}B)$, $\{v_i\}_i \subseteq E(\mathtt{t}B) - \{q\}$, and $\{w_j\}_j \subseteq E(\mathtt{t}C)$. Then

$$A \underset{[e]}{\square} B \;=\; X \underset{[e]}{\circ} (B \bigcirc_{\rho_B^{-1} v_i} V_i) \underset{\rho_B^{-1} q}{\circ} Y \underset{[r]}{\circ} \mathsf{Y}_{\mathtt{t}C} \bigcirc_{[w_j]} Z_j,$$

and $\rho_{A \,\square_{[e]}\, B}[f] = \rho_{A \,\square_{[e]}\, B}[eqr] = [e] \cdot \rho_B^{-1}\, q \cdot [r]$. Thus,

$$(A \underset{[e]}{\square} B) \underset{[e]\cdot\rho_B^{-1} q\cdot[r]}{\square} C \;=\; X \underset{[e]}{\circ} (B \bigcirc_{\rho_B^{-1} v_i} V_i) \underset{\rho_B^{-1} q}{\circ} Y \underset{[r]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j,$$

and the reindexing $\rho_{(A \,\square_{[e]}\, B) \,\square_{\rho_{A \,\square_{[e]}\, B}[f]}\, C \;\rho_{A \,\square_{[e]}\, B}}$ is given by

$$\begin{array}{llll}
[p] \in B^\bullet & \longmapsto & [ep] & \longmapsto \quad [ep] \\
 & & [p] \in C^\bullet & \longmapsto \quad [e] \cdot \rho_B^{-1}\, q \cdot [rp] \\
[ev_i s] & \longmapsto & [e] \cdot \rho_B^{-1}\, v_i \cdot [s] & \longmapsto \quad [e] \cdot \rho_B^{-1}\, v_i \cdot [s] \\
[eqs] \mathbin{\overline{+}} [f] & \longmapsto & [e] \cdot \rho_B^{-1}\, q \cdot [s] & \longmapsto \quad [e] \cdot \rho_B^{-1}\, q \cdot [s] \\
[f w_j s] & \longmapsto & [e] \cdot \rho_B^{-1}\, q \cdot [r] \cdot [w_j s] & \longmapsto \quad [e] \cdot \rho_B^{-1}\, q \cdot [r] \cdot \rho_C^{-1}\, w_j \cdot [s] \\
[p] \in A^\bullet \text{ n.a.a.} & \longmapsto & [p] & \longmapsto \quad [p]
\end{array}$$

(n.a.a. is an acronym by "not as above"). On the other hand, we have

$$A \underset{[f]}{\square} C \;=\; X \underset{[e]}{\circ} (\mathsf{Y}_{\mathtt{t}B} \bigcirc_{[v_i]} V_i) \underset{[q]}{\circ} Y \underset{[r]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j.$$

The reindexing gives $\rho_{A \,\square_{[f]}\, C}[e] = [e]$, and

$$\begin{aligned}
(A \underset{[f]}{\square} C) \underset{[e]}{\square} B \;&=\; X \underset{[e]}{\circ} (B \bigcirc_{\rho_B^{-1} v_i} V_i) \underset{\rho_B^{-1} q}{\circ} Y \underset{[r]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j \\
&=\; (A \underset{[e]}{\square} B) \underset{[e]\cdot\rho_B^{-1} q\cdot[r]}{\square} C.
\end{aligned}$$

The reindexing $\rho_{(A \,\square_{[f]}\, C) \,\square_{[e]}\, B \;\rho_{A \,\square_{[f]}\, C}}$ is given by

$$\begin{array}{lcllcl}
 & & [p] \in B^\bullet & \longmapsto & [ep] \\
[p] \in C^\bullet & \longmapsto & [fp] & \longmapsto & [e] \cdot \rho_B^{-1}\, q \cdot [rp] \\
[eqs] \fatsemi [f] & \longmapsto & [eqs] & \longmapsto & [e] \cdot \rho_B^{-1}\, q \cdot [s] \\
[ev_i s] & \longmapsto & [ev_i s] & \longmapsto & [e] \cdot \rho_B^{-1}\, v_i \cdot [s] \\
[f w_j s] & \longmapsto & [f] \cdot \rho_C^{-1}\, w_j \cdot [s] & \longmapsto & [e] \cdot \rho_B^{-1}\, q \cdot [r] \cdot \rho_C^{-1}\, w_j \cdot [s] \\
[p] \in A^\bullet \text{ n.a.a.} & \longmapsto & [p] & \longmapsto & [p]
\end{array}$$

We see that the square equation (A.2) commutes in the case $[e] \sqsubseteq [f]$.

(2) Assume $[e]$ and $[f]$ are $\sqsubseteq$-incomparable, and write A as

$$A \;=\; (X \underset{[e]}{\circ} \mathsf{Y}_{\mathsf{t}B} \bigcirc_{[v_i]} Y_i) \underset{[f]}{\circ} \mathsf{Y}_{\mathsf{t}C} \bigcirc_{[w_j]} Z_j.$$

Then

$$A \underset{[e]}{\square} B \;=\; (X \underset{[e]}{\circ} B \bigcirc_{\rho_B^{-1} v_i} Y_i) \underset{[f]}{\circ} \mathsf{Y}_{\mathsf{t}C} \bigcirc_{[w_j]} Z_j,$$

the reindexing gives $\rho_{A \square_{[e]} B}[f] = [f]$,

$$(A \underset{[e]}{\square} B) \underset{[f]}{\square} C \;=\; (X \underset{[e]}{\circ} B \bigcirc_{\rho_B^{-1} v_i} Y_i) \underset{[f]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j,$$

and the complete reindexing $\rho_{(A \square_{[e]} B) \square_{[f]} C}\, \rho_{A \square_{[e]} B}$ is given by

$$\begin{array}{lcllcl}
[p] \in B^\bullet & \longmapsto & [ep] & \longmapsto & [ep] \\
 & & [p] \in C^\bullet & \longmapsto & [fp] \\
[ev_i s] & \longmapsto & [e] \cdot \rho_B^{-1}\, v_i \cdot [s] & \longmapsto & [e] \cdot \rho_B^{-1}\, v_i \cdot [s] \\
[f w_j s] & \longmapsto & [f w_j s] & \longmapsto & [f] \cdot \rho_C^{-1}\, w_j \cdot [s] \\
[p] \in A^\bullet \text{ n.a.a.} & \longmapsto & [p] & \longmapsto & [p]
\end{array}$$

On the other hand,

$$A \underset{[f]}{\square} C \;=\; (X \underset{[e]}{\circ} \mathsf{Y}_{\mathsf{t}B} \bigcirc_{[v_i]} Y_i) \underset{[f]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j,$$

we have $\rho_{A \square_{[f]} C}[e] = [e]$,

$$\begin{aligned}
(A \underset{[f]}{\square} C) \underset{[e]}{\square} B \;&=\; (X \underset{[e]}{\circ} B \bigcirc_{\rho_B^{-1} v_i} Y_i) \underset{[f]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} Z_j \\
&=\; (A \underset{[e]}{\square} B) \underset{[f]}{\square} C,
\end{aligned}$$

and further

$$\begin{array}{lcllcl}
 & & [p] \in B^\bullet & \longmapsto & [ep] \\
[p] \in C^\bullet & \longmapsto & [fp] & \longmapsto & [fp] \\
[ev_i s] & \longmapsto & [ev_i s] & \longmapsto & [e] \cdot \rho_B^{-1}\, v_i \cdot [s] \\
[f w_j s] & \longmapsto & [f] \cdot \rho_C^{-1}\, w_j \cdot [s] & \longmapsto & [f] \cdot \rho_C^{-1}\, w_j \cdot [s] \\
[p] \in A^\bullet \text{ n.a.a.} & \longmapsto & [p] & \longmapsto & [p]
\end{array}$$

so that the square equation (A.2) commutes in the case where the addresses $[e]$ and $[f]$ are $\sqsubseteq$-incomparable too. Finally, the monad structure of M^+ satisfies condition **(Disjoint multiplication)** of definition A.1. □

Proof of **(Nested multiplication)**. Let $A, B, C \in \operatorname{tr} M$, $[e] \in A^\bullet$, $[f] \in B^\bullet$, such that $A \square_{[e]} B$ and $B \square_{[f]} C$ are admissible. Write A and B as:

$$A \;=\; (X \underset{[e]}{\circ} \mathsf{Y}_{\mathsf{t}B} \bigcirc_{[v_i]} Y_i), \qquad B \;=\; Z \underset{[f]}{\circ} \mathsf{Y}_{\mathsf{t}C} \bigcirc_{[w_j]} T_j.$$

Then,

$$A \underset{[e]}{\square} B = X \underset{[e]}{\circ} B \bigcirc_{\rho_B^{-1} v_i} Y_i,$$

we have $\rho_{A \square_{[e]} B}[f] = [ef]$, and

$$(A \underset{[e]}{\square} B) \underset{[ef]}{\square} C = X \underset{[e]}{\circ} (Z \underset{[f]}{\circ} \mathsf{Y}_{\mathsf{t}C} \bigcirc_{[w_j]} T_j) \bigcirc_{\alpha(\rho_B^{-1} v_i)} Y_i,$$

where

$$\alpha(\rho_B^{-1} v_i) = \begin{cases} [f] \cdot \rho_C^{-1} w_j \cdot [r] & \text{if } \rho_B^{-1} v_i \text{ of the form } [f w_j r], \\ \rho_B^{-1} v_i & \text{otherwise.} \end{cases}$$

Remark that $\alpha(\rho_B^{-1} v_i) = \rho_{B \square_{[f]} C}^{-1} v_i$. The reindexing $\rho_{(A \square_{[e]} B) \square_{[ef]} C} \rho_{A \square_{[e]} B}$ is given by:

$$\begin{array}{lllll} & [p] \in C^\bullet & \longmapsto & [efp] \\ [f w_j r] \in B^\bullet & \longmapsto & [ef w_j r] & \longmapsto & [ef] \cdot \rho_C^{-1} w_j \cdot [r] \\ [p] \in B^\bullet, [f] \not\equiv [p] & \longmapsto & [ep] & \longmapsto & [ep] \\ [e v_i r] \in A^\bullet & \longmapsto & [e] \cdot \rho_B^{-1} v_i \cdot [r] & \longmapsto & [e] \cdot \rho_{B \square_{[f]} C}^{-1} v_i \cdot [r] \end{array}$$

On the other hand, we have

$$\begin{aligned} B \underset{[f]}{\square} C &= Z \underset{[f]}{\circ} C \bigcirc_{\rho_C^{-1} w_j} T_j, \\ A \underset{[e]}{\square} (B \underset{[f]}{\square} C) &= X \underset{[e]}{\circ} (Z \underset{[f]}{\circ} \mathsf{Y}_{\mathsf{t}C} \bigcirc_{[w_j]} T_j) \bigcirc_{\rho_{B \square_{[f]} C}^{-1} v_i} Y_i \\ &= (A \underset{[e]}{\square} B) \underset{[ef]}{\square} C \end{aligned}$$

and the reindexing is given by

$$\begin{array}{lllll} [p] \in C^\bullet & \longmapsto & [fp] & \longmapsto & [efp] \\ [f w_j r] \in B^\bullet & \longmapsto & [f] \cdot \rho_C^{-1} w_j \cdot [r] & \longmapsto & [ef] \cdot \rho_C^{-1} w_j \cdot [r] \\ [p] \in B^\bullet, [f] \not\equiv [p] & \longmapsto & [p] & \longmapsto & [ep] \\ & & [e v_i r] \in A^\bullet & \longmapsto & [e] \cdot \rho_{B \square_{[f]} C}^{-1} v_i \cdot [r] \end{array}$$

We thus see that the square equation (A.3) commutes, and that the monad structure of M^+ satisfies condition **(Nested multiplication)**. □

This completes the proof of theorem A.14, endowing M^+ with a canonical monad structure, whose partial law is given by substitution of M-trees.

References

[1] John C. Baez and James Dolan. Higher-dimensional algebra. III. n-categories and the algebra of opetopes. *Advances in Mathematics*, 135(2):145–206, 1998.

[2] Krzysztof Bar, Aleks Kissinger, and Jamie Vicary. Globular: an online proof assistant for higher-dimensional rewriting. *arXiv e-prints*, page arXiv:1612.01093, December 2016.

[3] Eugenia Cheng. The category of opetopes and the category of opetopic sets. *Theory and Applications of Categories*, 11:No. 16, 353–374, 2003.

[4] Eric Finster. Opetopic.net. http://opetopic.net, May 2016.

[5] Eric Finster and Samuel Mimram. A Type-Theoretical Definition of Weak ω-Categories. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017.

[6] Peter Gabriel and Friedrich Ulmer. *Lokal präsentierbare Kategorien.* Lecture Notes in Mathematics, Vol. 221. Springer-Verlag, Berlin-New York, 1971.

[7] Nicola Gambino and Joachim Kock. Polynomial functors and polynomial monads. *Mathematical Proceedings of the Cambridge Philosophical Society*, 154(1):153–192, 2013.

[8] David Gepner, Rune Haugseng, and Joachim Kock. ∞-Operads as Analytic Monads. *International Mathematics Research Notices*, 04 2021. rnaa332.

[9] Victor Harnik, Michael Makkai, and Marek Zawadowski. Computads and multitopic sets. arXiv:0811.3215 [math.CT], 2008.

[10] Claudio Hermida, Michael Makkai, and John Power. On weak higher dimensional categories. I. 1. *Journal of Pure and Applied Algebra*, 154(1-3):221–246, 2000. Category theory and its applications (Montreal, QC, 1997).

[11] Claudio Hermida, Michael Makkai, and John Power. On weak higher-dimensional categories. I. 3. *Journal of Pure and Applied Algebra*, 166(1-2):83–104, 2002.

[12] Cédric Ho Thanh. The equivalence between opetopic sets and many-to-one polygraphs. arXiv:1806.08645 [math.CT], 2018.

[13] Cédric Ho Thanh. opetopy. https://github.com/altaris/opetopy, April 2018.

[14] Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and logics of computation (Cambridge, 1995)*, volume 14 of *Publ. Newton Inst.*, pages 79–130. Cambridge Univ. Press, Cambridge, 1997.

[15] Joachim Kock. Polynomial functors and trees. *International Mathematics Research Notices*, 2011(3):609–673, January 2011.

[16] Joachim Kock, André Joyal, Michael Batanin, and Jean-François Mascari. Polynomial functors and opetopes. *Advances in Mathematics*, 224(6):2690–2737, 2010.

[17] Stephen Lack and John Power. Gabriel-Ulmer duality and Lawvere theories enriched over a general base. *Journal of Functional Programming*, 19(3-4):265–286, 2009.

[18] Tom Leinster. *Higher Operads, Higher Categories*. Cambridge University Press, 2004.

[19] Jean-Louis Loday and Bruno Vallette. *Algebraic operads*, volume 346. Springer Science & Business Media, 2012.

[20] Michael Makkai. First order logic with dependent sorts, with applications to category theory. Available at https://www.math.mcgill.ca/makkai/folds/foldsinpdf/FOLDS.pdf, November 1995.