From Cubes to Twisted Cubes via Graph Morphisms in Type Theory

Gun Pinyo
School of Computer Science, University of Nottingham, UK
gunpinyo@gmail.com

Nicolai Kraus
School of Computer Science, University of Nottingham, UK
https://nicolaikraus.github.io/
nicolai.kraus@nottingham.ac.uk

Abstract

Cube categories are used to encode higher-dimensional categorical structures. They have recently gained significant attention in the community of homotopy type theory and univalent foundations, where types carry the structure of higher groupoids. Bezem, Coquand, and Huber [8] have presented a constructive model of univalence using a specific cube category, which we call the *BCH cube category*.

The higher categories encoded with the BCH cube category have the property that all morphisms are invertible, mirroring the fact that equality is symmetric. This might not always be desirable: the field of *directed type theory* considers a notion of equality that is not necessarily invertible.

This motivates us to suggest a category of *twisted cubes* which avoids built-in invertibility. Our strategy is to first develop several alternative (but equivalent) presentations of the BCH cube category using morphisms between suitably defined graphs. Starting from there, a minor modification allows us to define our category of twisted cubes. We prove several first results about this category, and our work suggests that twisted cubes combine properties of cubes with properties of globes and simplices (tetrahedra).

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases homotopy type theory, cubical sets, directed equality, graph morphisms

Digital Object Identifier 10.4230/LIPIcs.TYPES.2019.5

Related Version This paper is also available at https://arxiv.org/abs/1902.10820.

Funding *Nicolai Kraus*: The Royal Society, grant No. URF\R1\191055.

Acknowledgements We would like to thank Paolo Capriotti and Jakob von Raumer. Both offered many suggestions during fruitful exchanges. In particular, the initial observation on which Theorem 21 is based was suggested by them, and the idea of considering graph morphisms was found in one of our many interesting discussions. We are also grateful to the participants of TYPES'19 in Oslo and the summer school on HTT/UF in Leeds. We thank in particular Emily Riehl, Christian Sattler, and Steve Awodey for their help and their comments. Special thanks go to Andreas Nuyts, who has pointed out a mistake in an earlier draft of this paper, and to the anonymous reviewers for their careful reading and comments.

1 Introduction and Motivation

A *cube category* is a category whose objects are (or represent) finite-dimensional cubes, and whose morphisms are mappings of some sort between these cubes. There are many different cube categories [1, 5, 8, 9, 20], and they are used to encode higher categorical structures.

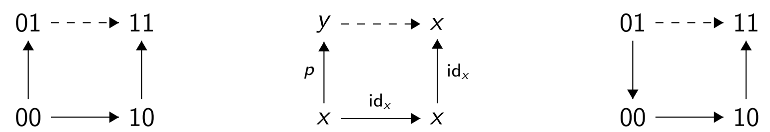


Figure 1 Kan-filling condition of a 2-cube (left), a proof of invertibility introduced by the Kan-filling condition (middle), and how to remove such the invertibility (right).

Homotopy type theory [28] is a variation of Martin-Löf's intensional type theory. The characteristic and novel view adapted in homotopy type theory is that types carry the structure of higher categories, or, to be precise, higher groupoids (i.e. all morphisms are invertible). This view supports Voevodsky's *univalence principle* which should be seen as a central concept of homotopy type theory. The first model of such a type theory, given by Voevodsky [29] (see also the presentation by Kapulkin and Lumsdaine [16]), uses *simplicial sets*. However, it is still an open question how simplicial sets can be used to build a *constructive* model of type theory with univalent universes [13]. Using *cubical sets*, this has been achieved by Bezem, Coquand, and Huber [8]. Starting from there, cubes have gathered a lot of attention in the type theory community, leading to various *cubical type theories* which have univalence not as an axiom but as a built-in derivable principle [3, 6, 12, 23]. Many different cube categories have been considered in this context.

The important cube category used by Bezem, Coquand, and Huber [8] (from now on referred to as the *BCH cube category*) uses finite sets of variable names as objects, and a morphism from a set I to a set J is a function $f : I \to J \cup \{0, 1\}$ which is "injective on the left part", i.e. $f(i_1) = f(i_2) = j$ with $j : J$ implies $i_1 = i_2$. One goal of this paper is to develop several alternative presentations of this category, mainly using graph morphisms. We have two main motivations to do this. The first is that, as we hope, our alternative and intuitive (but equivalent) definitions enable new views on the category and facilitate the discovery of further observations. The second motivation is that a minor change in the definition will allow us to construct a new cube category, the *twisted cubes* from the title. We will come back to this in a moment.

The standard way to create models (of both higher categories and type theories) using simplicial or cubical index categories is to take presheaves and equip them with certain *Kan-filling conditions*. These filling conditions entail composition of morphisms as well as associativity and all higher coherence laws that one needs. A typical such Kan-filling condition for the 2-cube[1], as shown on the left of Figure 1, says that, given the "partial square" of three solid edges on the right, one can always find the dashed edge (together with an actual filler for the square).

One important observation here is that, in the case of the BCH cube category and other cube categories, invertibility of morphisms is built-in. Consider the partial square, as shown on the middle of Figure 1, where two of the three solid edges are identities and the third is an actual non-trivial morphism (or equality) p from x to y. Using the Kan filling operation described above, we get a morphism from y to x, which serves as the inverse of p.

The invertibility of morphisms is useful for most forms of type theory, where equaliy is symmetric. This however is not always the case, cf. the proposals for *directed type theories* by Licata and Harper [18], Nuyts [22], Riehl and Shulman [25], North [21], and others. Their

[1] While Bezem, Coquand, and Huber [8] define their index category to have finite sets of variables as objects, it is possible to simply use natural numbers as objects. The *n-cube*, or *n-dimensional cube*, is then the object of the presheaf category that is represented by the object n of the index category.

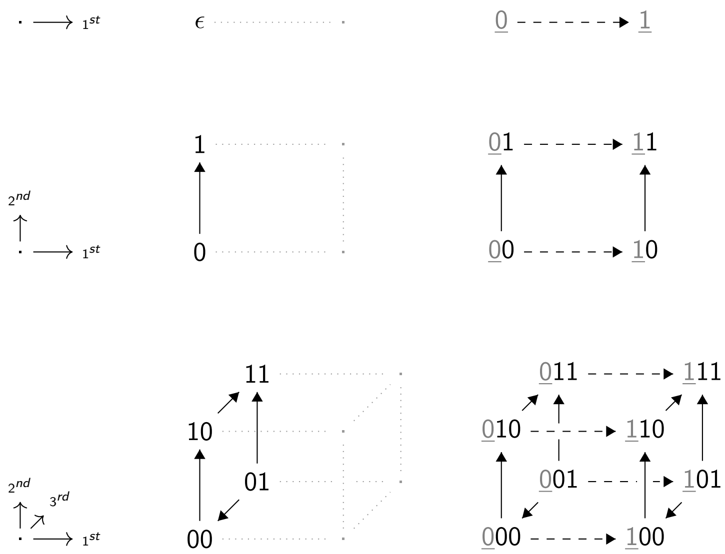


Figure 2 An illustration of the *thickening-and-twisting* process of the twisted n-cube for $1 \leqslant n \leqslant 3$. The process expands the twisted $(n-1)$-cube (left column) along the new dimension (middle column) and reverse all other dimensions at the starting point of the new dimension (right column).

aim is to generalise type theory by replacing *(higher) groupoids* by general *(higher) categories*. In a nutshell, this means that "equality" (or whatever takes the place of equality) is not necessarily invertible.

We think that a very valuable long-term goal would be to make the connection of directed type theories with cubical type theories and create some sort of *directed cubical type theory*. This is at the moment certainly out of reach, and we do not know how such a type theory could be built. Nevertheless, it motivates us to explore variations of the BCH cube category which do not have the described built-in equality.

To avoid invertibility, we "twist" the left-most edge of the 2-dimensional cube, as shown on the right of Figure 1, to ensure that the construction from before becomes impossible. This might seems artificial and specific to the 2-dimensional case but by using our graph morphisms that we develop for the BCH cube category, it becomes very easy to define the twisting version for cubes of all dimensions.

To construct a twisted n-cube from a twisted $(n-1)$-cube, we first expand the original cube along a new dimension (we call this *thickening*): this is same as constructing a standard n-cube form a standard $(n-1)$-cube, which is just a construction of its *cylinder object*. We then reverse all dimensions at the starting point of the new dimension (we call this *twisting*). Figure 2 illustrates this *thickening-and-twisting* process up to dimension 3, where the existing dimensions are shifted by one in order to allow the new dimension to be the first dimension.

One important property of standard cubes which twisted cubes retain is that every face of a [twisted] n-cube is a [twisted] $(n-1)$-cube. An interesting example is the case $n = 3$: In order to construct a twisted 3-cube, we thicken the twisted 2-cube as illustrate in Figure 2 where the left and the right face are already twisted 2-cubes, while the rest are thickened 1-cubes. The right face is unaffected during the twisting, but the left face is reversed entirely. Nevertheless, it is still a 2-cube (just flipped backwards).

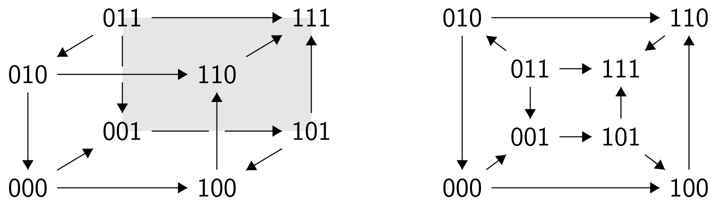


Figure 3 The 3-dimensional twisted cube using parallel and perspective projections. On the left, the lid (i.e. the last face which can be recovered by filling) is shaded. On the right, this face is the small middle square. The lid can be seen as the composite of the other faces.

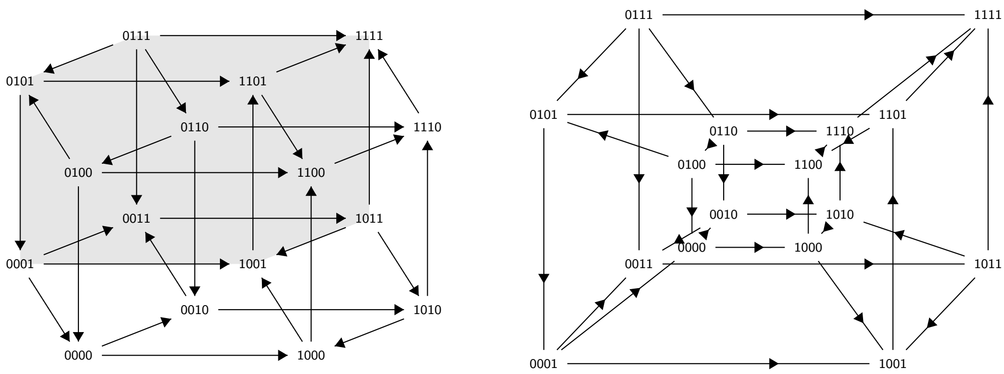


Figure 4 The 4-dimensional twisted cube using parallel and perspective projections. The lid is shadowed on the left. It is the biggest cube on the right.

Twisted cubes do not only remove the discussed source of invertibility, but they also change the way we view composition of morphisms. The filling of a "standard" square can be interpreted as saying that the composition of two edges equals the composition of the other two edges, and if we want to see the lid as the composite of the three other edges, then one has to be inverted. In contrast, in the twisted square, the lid can be seen directly as the single composite of the three other edges. The right half of Figure 3 shows the projection of the twisted 3-cube, and the smallest square (011, 001, 101, 111) is the lid. As for the square, this lid should be seen as the composite of the other (here five) faces. Intuitively, one starts with the biggest square, composes it with the top and the bottom squares, then with the left and the right square, and thus arrives at the smallest square. Figure 4 shows the analogous situation for the 4-dimensional twisted cube, where one starts with the inner 3-cube, then extends to the front and the back, to the top and the bottom, and finally to the left and the right.

The "twisting" pattern also appears in the *twisted arrow category* [17], also known as the *category of factorisations* [7]. However, it is unclear how to generalise this idea to more than squares; it has been developed to solve a different problem.

In the main body of the paper, we first introduce the framework of graph morphisms for standard (non-twisted) cubes. We consider the properties of meet/join and dimension preservation of graph morphisms, and conclude that both of these are suitable refinements to ensure that the category of graph morphisms matches the BCH cube category. The proof of this is the main result of Section 2. We use this development to introduce and examine *twisted cubes* in Section 3. We will see that they have many characteristic properties that standard cubes are lacking. Some of them, such as a Hamiltonian path through the cube and the fact that vertices are totally ordered, are familiar from simplicial structures but not from cubical ones. Another interesting feature, neither familiar from cubical nor from simplicial but from globular structures, is that surjective maps are unique (i.e. there is only one way to degenerate a twisted cube). These and other observations allow us to define a further representation of the category of twisted cubes which does not make use of graphs.

Setting. We use a standard version of Martin-Löf's dependent type theory as our meta-language. We assume function extensionality, but we do not require other axioms or features since we mostly work with finite sets, which are extremely well-behaved by default. In particular, it does not matter for us whether UIP/Axiom K is assumed or not, and the development would be identical in extensional dependent type theory.

Summary of Contributions. Our main contributions are as follows:

- We give several alternative but equivalent presentations of the BCH cube category.
- We introduce *twisted cubes*, a variation of the BCH cube category which allows for filling conditions without built-in invertibility.
- We show several results about twisted cubes. These include connections to simplices (a unique Humiliation path and the property of being a Reedy category) and to globes (unique surjective maps and degeneracies).

2 A Standard Cube Category

In this section, we discuss various representations of the cube category $\square_{\mathsf{BCH}}$. This category was used by Bezem, Coquand, and Huber to present a constructive model of univalence [8]. In Section 3, we will see how minimal modifications lead to a category of twisted cubes.

Keeping in mind that we use type theory as the language in which the results are presented (i.e. as our meta-theory), we use the following notations: $\mathbb{N}$ are the natural numbers, including 0. For $n : \mathbb{N}$, the set $\underline{n}$ is the finite set with elements $\{0, 1, \ldots, n-1\}$. In particular, $\underline{2}$ is the set of booleans. As usual, $\underline{n}^{\underline{m}}$ is simply the function set $\underline{m} \to \underline{n}$. We denote elements of $\underline{2}^{\underline{n}}$ by binary sequences as in $0 \cdot 1 \cdot 1 \cdot 0$. This means such a function f is denoted by $f(0) \cdot f(1) \cdot f(2) \ldots f(n-1)$. If there is no risk of confusion, we omit the $\cdot$ and simply use juxtaposition as in 0110.

In several situations, we want to consider a type of functions into a coproduct which is injective "on the *left* part of the codomain". To make this precise, we introduce a notation:

▶ **Definition 1** ($\xhookrightarrow{\mathsf{left}}$)**.** *Assume A, B, and C are given types. For a function $f : A \to (B + C)$, we say that f is* injective on the left part *if*

$$\mathsf{left\text{-}inj}(f) :\equiv \Pi(x, y : A, z : B).(f(x) = \mathsf{inl}(z)) \to (f(y) = \mathsf{inl}(z)) \to x = y. \tag{1}$$

We write the type of functions which are injective on the left part as

$$(A \xhookrightarrow{\mathsf{left}} B + C) :\equiv \Sigma(f : A \to (B + C)).\mathsf{left\text{-}inj}(f). \tag{2}$$

In the next lemma, a function $f : A \to B + \underline{1}$ is called a *partial function*, with $\underline{1}$ being the "undefined" part.[2] The following simple but useful (and well-known) result will be necessary. It could be formulated in higher generality, but a version which is sufficient for us is this:

▶ **Lemma 2.** *Given $m, n : \mathbb{N}$, injective partial functions from $\underline{m}$ to $\underline{n}$ are in bijection with injective partial functions from $\underline{n}$ to $\underline{m}$. In other words, we have an equivalence*

$$\left(\underline{m} \xhookrightarrow{\mathsf{left}} \underline{n} + \underline{1}\right) \simeq \left(\underline{n} \xhookrightarrow{\mathsf{left}} \underline{m} + \underline{1}\right). \tag{3}$$

Proof. The equivalence can be constructed directly. Given an $f : \underline{m} \xhookrightarrow{\mathsf{left}} \underline{n} + \underline{1}$, we have to construct a function $g : \underline{n} \xhookrightarrow{\mathsf{left}} \underline{m} + \underline{1}$. For $i : \underline{n}$, we can decide whether there is a k such that $f(k) = \mathsf{inl}(i)$. If so, then this k is unique due to injectivity, and we set $g(i) :\equiv \mathsf{inl}(k)$; otherwise, we set $g(i) :\equiv \mathsf{inr}(0)$. Checking that this is an equivalence is routine. ◀

The presentation of the cube category in question that we start with is the one given by Bezem, Coquand, and Huber [8] (which is the same as in Huber's PhD thesis [15]). Since it is sufficient for our purposes, we use a skeletal variation: our objects are not finite sets but rather natural numbers.

▶ **Definition 3** (category $\square_{\mathsf{BCH}}$ [8, 15]). *The category $\square_{\mathsf{BCH}}$ has natural numbers as objects and, for $m, n : \mathbb{N}$, a morphism in $\square_{\mathsf{BCH}}(m, n)$ is a function $f : \underline{m} \to \underline{n} + \underline{2}$ which is injective on the $\underline{n}$-part. In type-theoretic notation:*

$$\mathsf{obj}(\square_{\mathsf{BCH}}) :\equiv \mathbb{N} \qquad\qquad \square_{\mathsf{BCH}}(m, n) :\equiv \underline{m} \xhookrightarrow{\mathsf{left}} \underline{n} + \underline{2} \tag{4}$$

Composition $g \circ f$ is defined to be the set-theoretic composition $(g + \mathsf{id}_2) \circ f$.

What we will need is the opposite of this category, $\square^{\mathsf{op}}_{\mathsf{BCH}}$. While the above definition is short and abstract, a description close to the intuitive idea of cubes is helpful for our later developments. Let us consider *graphs* $G = (V, E)$ of nodes (vertices) and edges, where V is a set with decidable equality and E is a subset of $V \times V$. A standard way to implement this is to let E be a family of "mere propositions"[3], indexed twice over V. However, we write $(s, t) : E$ for $E(s, t)$ and assume that E is given in the "total space" formulation. Furthermore, in our cases E will always be a *decidable* subset.

E being a subset means that our graphs do *not* have multiple parallel edges, i.e. for any pair of vertices, there is at most one edge between them, and it is decidable whether there is an edge between two given vertices.

Given a graph, we construct a new graph as follows. Note that the "total space" of the edges of the new graph is $E + E + V$, but in order to make clear which vertices these new edges connect, we use "set theory style" notation:

▶ **Definition 4.** *Given $G = (V, E)$, the* graph-prism *of G, denoted as* $\mathsf{prism}\,(G) :\equiv (\mathsf{prism}\,(V), \mathsf{prism}\,(E))$ *is another graph where*

$$\mathsf{prism}\,(V) :\equiv \underline{2} \times V \tag{5}$$

$$\mathsf{prism}\,(E) :\equiv \{\,(\,(0,\ s),\ (0, t)\,)\ \mid\ (s, t) : E\} \tag{6}$$

$$\cup \{\,(\,(1,\ s),\ (1, t)\,)\ \mid\ (s, t) : E\} \tag{7}$$

$$\cup \{\,(\,(0,\ v),\ (1, v)\,)\ \mid\ v : V\}. \tag{8}$$

[2] Technically, these are of course only the partial functions from A to B with decidable support. Since we only work with finite types, it is not surprising that we only need to consider the decidable case.

[3] Recall that a *mere proposition*, or a *subsingleton*, is a type with at most one element.

This allows us to define the standard cube as a graph:[4]

▶ **Definition 5.** *Given* $n : \mathbb{N}$*, the standard cube* C_n *is defined as follows:*

$$C_0 :\equiv (\underline{1}, \{(0,0)\}) \qquad C_{n+1} :\equiv \mathsf{prism}\,(C_n) \tag{9}$$

Another way of defining C_n, without recursion, is the following. Here, we give the "total space" of edges $\mathsf{edges}(C_n)$ together with functions $\mathsf{src}, \mathsf{trg} : \mathsf{edges}(C_n) \to \mathsf{nodes}(C_n)$:

▶ **Definition 6.** *In the following, our convention is that* $\underline{-1}$ *is empty (i.e. the same as* $\underline{0}$*):*

$$\mathsf{nodes}(C_n) :\equiv \underline{2^n} \tag{10}$$
$$\mathsf{edges}(C_n) :\equiv \underline{2^n} + (\underline{n} \times \underline{2^{n-1}}) \tag{11}$$
$$\mathsf{src}(\mathsf{inl}(v)) :\equiv \mathsf{trg}(\mathsf{inl}(v)) :\equiv v \tag{12}$$
$$\mathsf{src}(\mathsf{inr}(i, x_0x_1 \ldots x_{n-2})) :\equiv x_0x_1 \ldots x_{i-1}0x_i \ldots x_{n-2} \tag{13}$$
$$\mathsf{trg}(\mathsf{inr}(i, x_0x_1 \ldots x_{n-2})) :\equiv x_0x_1 \ldots x_{i-1}1x_i \ldots x_{n-2} \tag{14}$$

The number of total edges in (11) comes from the following calculation. We have n dimension, thus 2^n nodes, which come with self-loops giving rise to the summand $\underline{2^n}$. For ever node, we further have an edge in each dimension. Avoiding double counting, this gives the summand $\underline{n} \times \underline{2^{n-1}}$. Figure 5 shows drawings for C_0 to C_3.

▶ **Lemma 7.** *Definition 5 and Definition 6 define isomorphic graph structures.* ◀

This observation allows us to use whichever is more convenient in any given situation.

A *graph morphism* from $G = (V, E)$ to $G' = (V', E')$ is, as usual, a function between the node types which preserves the edges:

$$\mathsf{grp\text{-}hom}\,((V, E), (V', E')) :\equiv \Sigma(f : V \to V').\Pi(v_0, v_1 : V).E(v_0, v_1) \to E'(f(v_0), f(v_1)) \tag{15}$$

We can now consider the following category:

▶ **Definition 8** (category $\square_{\mathsf{grp}}$)**.** *The category* $\square_{\mathsf{grp}}$ *has natural numbers as objects. A morphism between* m *and* n *is a graph morphism from* C_m *to* C_n*, as in:*

$$\mathsf{obj}(\square_{\mathsf{grp}}) :\equiv \mathbb{N} \qquad \square_{\mathsf{grp}}(m, n) :\equiv \mathsf{grp\text{-}hom}\,(C_m, C_n) \tag{16}$$

Composition is composition of graph morphisms.

[4] Most of graphs in this paper are reflexive graphs to support degeneracies as graph morphisms.

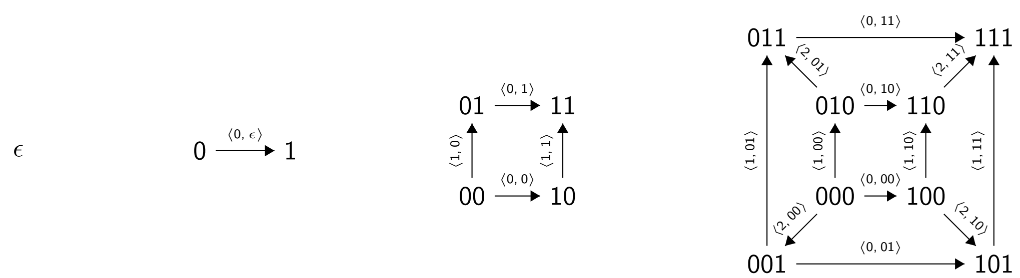


Figure 5 An illustration of C_n for $n \leqslant 3$. The labels on the vertices and edges are in accordance with (10) and (11). The identity loops are omitted. This allows us to unambiguously hide the constructor inr as well.

The category $\square_{\mathsf{grp}}$ has more morphisms than $\square_{\mathsf{BCH}}^{\mathsf{op}}$. One example would be the morphism in $\mathsf{grp\text{-}hom}\,(C_2, C_1)$ which maps the three nodes 00, 01, 10 all to 0 and 11 to 1. Another example is the morphism which maps 00 to 0, and 01, 10, 11 all to 1. Both of these morphisms do not have analogues in $\square_{\mathsf{BCH}}^{\mathsf{op}}$. In other words, $\square_{\mathsf{grp}}$ has *connections*. Since, in the current paper, we are looking for alternative definitions of the category $\square_{\mathsf{BCH}}^{\mathsf{op}}$, we refine the definition of the morphisms in $\square_{\mathsf{grp}}$ to resolve the mismatch. Let us formulate the following auxiliary definitions.

▶ **Definition 9** (free preorder on a graph)**.** *For a given graph $G = (V, E)$, we write $G^* = (V, E^*)$ for the free preorder generated by it. G^* has V as objects and, for $v, u : V$, we have $v \leqslant u$ if there is a chain of edges starting in v and ending in u.*

When talking about nodes in G, we borrow the notions of meet *(product) and* join *(coproduct) from preorders. If they exist in G^*, we write them as $v \sqcap u$ and $v \sqcup u$.*

It is easy to see that, in the case of C_n, all meets and joins exist and can be calculated directly: From the programming perspective, they correspond to the bitwise operators $'\&'$ and $'|'$. Thus, when talking about C_n, we can view $\sqcap$ and $\sqcup$ as actual functions calculating the binary meet and join:

$$\sqcap, \sqcup : V \times V \to V \tag{17}$$

Given a graph morphism $g : \mathsf{grp\text{-}hom}\,(C_m, C_n)$, it is easy to define what it means that it preserves binary meets resp. joins:

$$\mathsf{pres\text{-}meet}(g) :\equiv \Pi(u, v : \underline{2}^{m}).g(u \sqcap v) = g(u) \sqcap g(v) \tag{18}$$

$$\mathsf{pres\text{-}join}(g) :\equiv \Pi(u, v : \underline{2}^{m}).g(u \sqcup v) = g(u) \sqcup g(v) \tag{19}$$

Note that preserving meets and joins is a property (a "mere proposition") of morphisms. For general morphisms between graphs which might not have all meets or joins, the definition is more subtle but still straightforward; one can always define the property of *being a meet (join)* and then say that any vertex which has this property is mapped to one which also has it. We omit the precise type-theoretic formulation.

The two mentioned examples of morphisms which are "too much" in $\square_{\mathsf{grp}}$ do not preserve binary meets resp. joins.

▶ **Definition 10** (category $\square_{\mathsf{cont}}$)**.** *The category $\square_{\mathsf{cont}}$ has $\mathbb{N}$ as objects and, as morphisms, graph morphisms between standard cubes which preserve meets and joins (*cont *for* continuous*):*

$$\mathsf{obj}(\square_{\mathsf{cont}}) :\equiv \mathbb{N} \tag{20}$$

$$\square_{\mathsf{cont}}(m, n) :\equiv \Sigma(g : \mathsf{grp\text{-}hom}\,(C_m, C_n)).\mathsf{pres\text{-}meet}(g) \times \mathsf{pres\text{-}join}(g) \tag{21}$$

This gives us a category which is indeed equivalent (in fact isomorphic) to $\square_{\mathsf{BCH}}^{\mathsf{op}}$:

▶ **Theorem 11.** *The categories $\square_{\mathsf{BCH}}^{\mathsf{op}}$ and $\square_{\mathsf{cont}}$ are isomorphic. The isomorphism on the object part is the identity, i.e. the equivalence is given by a family e as in:*

$$e : \Pi(m, n : \mathbb{N}).\square_{\mathsf{BCH}}^{\mathsf{op}}(m, n) \simeq \square_{\mathsf{cont}}(m, n). \tag{22}$$

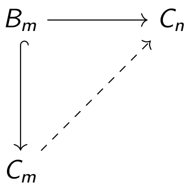
Before giving a proof, we formulate the following:

▶ **Lemma 12.** *Consider the full subgraph of C_n which has exactly $(n+1)$ vertices, namely the "origin" $00\ldots0$ and the "base vectors" which have exactly one 1. We call this subgraph B_n, where the B stands for "base", and it comes with the inclusion $i : B_n \hookrightarrow C_n$. For any m, "forgetting" the property of preserving the joins and composing with i as in*

$$\lambda g.i \circ (\mathsf{proj}_1(g)) : \ (\Sigma(g : \mathsf{grp\text{-}hom}\,(C_m, C_n)\,.\mathsf{pres\text{-}join}(g)) \ \to \ \mathsf{grp\text{-}hom}\,(B_m, C_n) \tag{23}$$

is an equivalence. Moreover, g preserves meets if and only if $i \circ (\mathsf{proj}_1(g))$ does.

Proof. The only binary joins that B_m has are trivial, so every morphism $\mathsf{grp\text{-}hom}\,(B_m, C_n)$ is join-preserving. Thus, the first claim of the lemma is that every such morphism can be extended in a unique way as shown in the diagram to the right. Every node of C_m which is not in B_m, i.e. every node which is not the origin or a base vector, can be written as a join of base vectors. Since we need to preserve joins, it is therefore determined where the node has to be sent to. The map defined in this way preserves all binary joins, and it preserves binary meets if and only if the input does. ◀



Proof of Theorem 11. We first give the overview of the argument as a chain of equivalences, then we justify each step [S1 – S5].

$$
\begin{aligned}
& \square_{\mathsf{cont}}(m,n) \\
& \equiv \Sigma(g : \mathsf{grp\text{-}hom}\,(C_m, C_n)).\mathsf{pres\text{-}meet}(g) \times \mathsf{pres\text{-}join}(g) \\
\text{[S1]} \;& \simeq \Sigma(g : \mathsf{grp\text{-}hom}\,(B_m, C_n)).\mathsf{pres\text{-}meet}(g) \\
\text{[S2]} \;& \simeq \Sigma(z : \underline{2}^{\underline{n}}, d : \underline{m} \xhookrightarrow{\mathsf{left}} \underline{n} + \underline{1}).\Pi(i : \underline{m}, j : \underline{n}).(d(i) = \mathsf{inl}(j)) \to (z(j) = 0) \\
\text{[S3]} \;& \simeq \Sigma(z : \underline{2}^{\underline{n}}, e : \underline{n} \xhookrightarrow{\mathsf{left}} \underline{m} + \underline{1}).\Pi(i : \underline{m}, j : \underline{n}).(e(j) = \mathsf{inl}(i)) \to (z(j) = 0) \\
\text{[S4]} \;& \simeq \Sigma(z : \underline{2}^{\underline{n}}, e : \underline{n} \to (\underline{m} + \underline{1})).\mathsf{left\text{-}inj}(e) \times \Pi(i : \underline{m}, j : \underline{n}).(e(j) = \mathsf{inl}(i)) \to (z(j) = 0) \\
\text{[S5]} \;& \simeq \Sigma\big(\alpha : \Pi(j : \underline{n}).\Sigma(e : \underline{m} + \underline{1}, z : \underline{2}).\Pi(i : \underline{m}).(e = \mathsf{inl}(i)) \to z = 0\big).\mathsf{left\text{-}inj}(\mathsf{proj}_1 \circ \alpha) \\
\text{[S6]} \;& \simeq \Sigma\big(\alpha : \Pi(j : \underline{n}).\underline{m} + \underline{2}\big).\mathsf{left\text{-}inj}(\alpha) \\
& \equiv \square^{\mathsf{op}}_{\mathsf{BCH}}(m,n)
\end{aligned}
$$

Step 1 holds by Lemma 12. Let us look at Step 2. Giving a graph homomorphism between B_m and C_n corresponds to choosing where the origin is mapped to, and choosing where each (non-trivial) edge of B_m is mapped to. For the origin, we use the component $z : \underline{2}^{\underline{n}}$. There are m non-trivial edges in B_m, and z is an endpoint (or starting point) of n non-trivial edges and one trivial edge in C_n. This gives us up to $\underline{m} \to \underline{n} + \underline{1}$ possible functions, but since we only consider meet-preserving morphisms, every function needs to be injective on the left part, leading to $d : \underline{m} \xhookrightarrow{\mathsf{left}} \underline{n} + \underline{1}$. Moreover, if $d(i) = \mathsf{inl}(j)$ for some i, j, then the image of the origin must be the *starting point* of the edge in dimension j, i.e. $z(j) = 0$. Step 3 is an application of Lemma 2 (it essentially swaps the roles of m and n). Step 4 only unfolds the definition of $\xhookrightarrow{\mathsf{left}}$.

In Step 5, the usual distributivity between Σ and Π (under the propositions-as-types view referred to as the "axiom of choice") is used: z, e, and the unnamed last component can all be seen as (dependent) functions with domain $\underline{n}$. The dependent function α combines them into a single dependent function with domain $\underline{n}$ and a codomain that consists of multiple components which, again, are called e, z, with the last one being unnamed. Only the component expressing the "injectivity on the left part"-property cannot be seen as a function in $\underline{n}$. In Step 6, we massage the codomain of α: We have $e : \underline{m} + \underline{1}$ and also $z : \underline{2}$, but the condition says that z is determined unless $e = \mathsf{inr}(0)$; thus, the type is equivalent to $\underline{m} + \underline{2}$.

We omit the calculation which shows that the constructed equivalence preserves composition of morphisms in the categories. ◀

In Section 3, we will switch from standard cubes to twisted cubes. The directions of some edges will be reversed. It is therefore an advantage to formulate a condition similar to the one about meets and joins without referring to the direction of edges. This is indeed possible:

▶ **Definition 13** (dimension preserving morphisms; category $\square_{\mathsf{dim}}$). *Given the standard cube C_n, where we use the non-recursive definition as in Definition 6, the* dimension *of an edge is defined as follows:*

$$\mathsf{dim} : \mathsf{edges}(C_n) \to \underline{n} + \underline{1} \qquad \mathsf{dim}(\mathsf{inl}(v)) :\equiv \mathsf{inr}(0) \tag{24}$$

$$\mathsf{dim}(\mathsf{inr}(i, x_0 \ldots x_{n-2}) :\equiv \mathsf{inl}(i) \tag{25}$$

We say that a morphism $f : \mathsf{grp\text{-}hom}\,(C_m, C_n)$ is dimension-preserving *if f maps edges of the same dimension to edges of the same dimension,*

$$\mathsf{dim\text{-}pres}(f) :\equiv \Pi(e_1, e_2 : \mathsf{edges}(C_n)).(\mathsf{dim}(e_1) = \mathsf{dim}(e_2)) \to (\mathsf{dim}(f(e_1)) = \mathsf{dim}(f(e_2))). \tag{26}$$

The category $\square_{\mathsf{dim}}$ makes use of these concepts:

$$\mathsf{obj}(\square_{\mathsf{dim}}) :\equiv \mathbb{N} \qquad \square_{\mathsf{dim}}(m, n) :\equiv \Sigma(g : \mathsf{grp\text{-}hom}\,(C_m, C_n)).\mathsf{dim\text{-}pres}(g) \tag{27}$$

As $\mathsf{pres\text{-}meet}(g)$ and $\mathsf{pres\text{-}join}(g)$, preserving the dimension as in (26) is a proposition in the sense of homotopy type theory (has at most one proof).

▶ Remark 14. For a graph morphism f as in the definition above, the following condition says that f is "injective on dimensions" (on the non-trivial part):

$$\mathsf{dim\text{-}inj}(f) :\equiv \Pi(e_1, e_2 : \mathsf{edges}(C_m), j : \underline{n}).(\mathsf{dim}(f(e_1)) = \mathsf{inl}(j) \times \mathsf{dim}(f(e_2)) = \mathsf{inl}(j))$$
$$\to (\mathsf{dim}(e_1) = \mathsf{dim}(e_2)).$$

However, note that this follows directly from $\mathsf{dim\text{-}pres}(f)$: Assume e_1, e_2 are edges such that $\mathsf{dim}(f(e_1))$ and $\mathsf{dim}(f(e_2))$ are equal and non-trivial. If e_1 and e_2 are not "parallel" (i.e. not in the same dimension), then we can find e_1' in the same dimension as e_1 such that e_1' and e_2 are adjacent (i.e. the endpoint of one is the starting point of the other). It is clear that $f(e_1')$ and $f(e_2)$ cannot go into the same non-trivial direction, since we can only go one step into a given direction before going back.

The connection to meet- and join-preserving is given by the following result:

▶ **Lemma 15.** *A morphism $f : \mathsf{grp\text{-}hom}\,(C_m, C_n)$ is join-and-meet-preserving exactly if it is dimension-preserving.*

Proof. This follows easily by going via morphisms $\mathsf{grp\text{-}hom}\,(B_m, C_n)$ as in Lemma 12. The graph B_m has exactly one edge for every non-trivial dimension, and the proof is analogous to the one of Lemma 12. ◀

▶ **Corollary 16** (Section summary). *The categories $\square_{\mathsf{BCH}}^{\mathsf{op}}$, $\square_{\mathsf{cont}}$, and $\square_{\mathsf{dim}}$ are isomorphic.* ◀

3 A Category of Twisted Cubes

As discussed in the introduction, we build on our framework of graph morphisms to define a category of *twisted cubes*. A variation of Definition 4 gives us these twisted cubes. The critical change can be seen in (29), which should be compared with (6):

▶ **Definition 17.** *Given a graph* $G = (V, E)$*, the* twisted graph-prism *of* G*, denoted as* $\mathsf{tw\text{-}prism}\,(G) :\equiv (\mathsf{tw\text{-}prism}\,(V),\ \mathsf{tw\text{-}prism}\,(E))$ *is the graph defined by*

$$\mathsf{tw\text{-}prism}\,(V) \ :\equiv \underline{2} \times V \tag{28}$$

$$\mathsf{tw\text{-}prism}\,(E) \ :\equiv \{\ (\ (0,\ t),\ (0, s)\ )\ \mid\ (s, t) : E\} \tag{29}$$

$$\cup \{\ (\ (1,\ s),\ (1, t)\ )\ \mid\ (s, t) : E\} \tag{30}$$

$$\cup \{\ (\ (0,\ v),\ (1, v)\ )\ \mid\ v : V\}. \tag{31}$$

We then define:

▶ **Definition 18.** *Given* $n : \mathbb{N}$*, the twisted cube* T_n *is defined as follows:*

$$T_0 \ :\equiv\ (\underline{1},\ \{(0, 0)\}) \qquad\qquad T_{n+1} \ :\equiv\ \mathsf{tw\text{-}prism}\,(T_n) \tag{32}$$

Alternatively, we can tweak Definition 5 to get a non-recursive definition. As before, the convention is that $\underline{-1}$ is empty.

▶ **Definition 19.** *The non-recursive definition of* T_n *is as follows:*

$$\mathsf{nodes}(T_n) \ :\equiv\ \underline{2^n} \tag{33}$$

$$\mathsf{edges}(T_n) \ :\equiv\ \underline{2^n} + (\underline{n} \times \underline{2^{n-1}}) \tag{34}$$

$$\mathsf{src}(\mathsf{inl}(v)) \ :\equiv\ \mathsf{trg}(\mathsf{inl}(v)) \ :\equiv\ v \tag{35}$$

$$\mathsf{src}(\mathsf{inr}(i, x_0 x_1 \ldots x_{n-2})) \ :\equiv\ x_0 x_1 \ldots x_{i-1} \cdot b \cdot x_i \ldots x_{n-2} \tag{36}$$

$$\mathsf{trg}(\mathsf{inr}(i, x_0 x_1 \ldots x_{n-2})) \ :\equiv\ x_0 x_1 \ldots x_{i-1} \cdot (1 - b) \cdot x_i \ldots x_{n-2} \tag{37}$$

where $b = 1$ *if the total number of zeros in* $x_0 x_1 \ldots x_{i-1}$ *is odd, and* $b = 0$ *otherwise.*

This means that an edge is reversed (compared to the standard cubes discussed before) exactly if the number of zeros in dimensions that come *before* the edge is odd (note that the condition talks about x_{i-1}, not x_{n-2}). The twisted cubes of dimension up to 3 are illustrated in Figure 6; see also Figures 3 and 4 in the introduction.

▶ **Lemma 20.** *Definition 18 and Definition 19 define isomorphic graph structures.* ◀

T_n has an interesting property that the standard cube C_n does not have: The induced preorder T_n^* on the vertices is a total order. This observation was originally suggested by Paolo Capriotti and Jakob von Raumer in a discussion with the first author of this paper. Note that this observation should not be misunderstood to mean that T_n itself is uninteresting. Its edges give it a unique structure, as visualised in Figure 7.

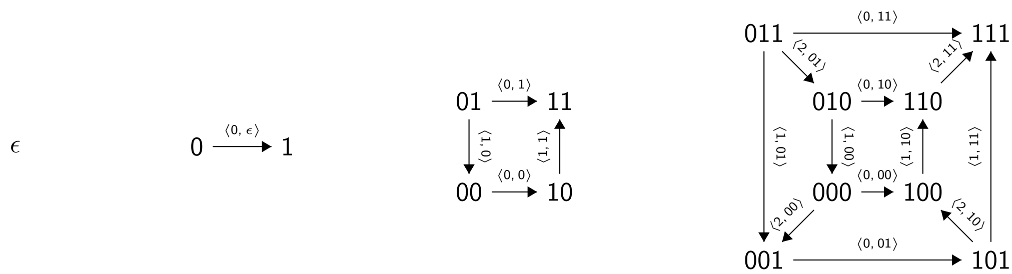


Figure 6 An illustration of T_n where $n \leqslant 3$.

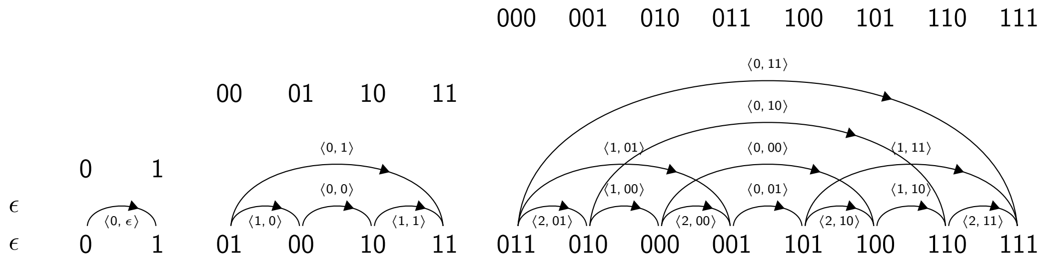


Figure 7 Linear drawings of the twisted cubes T_0, T_1, T_2, and T_3, demonstrating that the underlying preorders are total orders. The binary sequences on top are the values of g_n from the proof of Theorem 21. See also Remark 22.

The idea behind this result is that tw-prism preserves the property of having a preorder that is total. To elaborate on this, if G^* is a total order, then $(\text{tw-prism } G)^*$ consists of two copies of G^*, where the first copy is "turned around". One of the edges added in (31) links the largest node in the first copy to the smallest node in the second copy, thus every element of the second copy is larger than all the elements of the first copy. In other words, $(\text{tw-prism } G)^*$ is the *join* of the two copies.[5]

▶ **Theorem 21.** *For all $n : \mathbb{N}$, the preorder T_n^* is isomorphic to the total order $(\underline{2}^{\underline{n}}, <)$.*

Note that Theorem 21 is a property which one usually expects for simplicial structures, but not for cubical ones.

▶ Remark 22. There are two binary numbers for each node in Figure 7. The bottom one represents each node name according to Definition 19 whereas the top one represents the total order of T_3. It is impossible to unify these two binary numbers for $n \geqslant 2$ since, for each edge e, the numbers $\mathsf{src}(e)$ and $\mathsf{src}(e)$ only differ by (at most) one single bit by Definition 19, while incrementing a binary number can flip more than one bit.

Another related observation is that we can find a path from the smallest vertex to the largest vertex of T_n which respects the direction of the edges, and which visits each vertex exactly once. Recall that such a path is called a *Hamiltonian path*. We record this:

▶ **Theorem 23.** *For all $n : \mathbb{N}$, there is exactly one Hamiltonian path through T_{n+1}. This path contains exactly one edge in the first dimension (i.e. the one which is added when going from T_n to T_{n+1}). Moreover, this single edge in the new dimension connects the Hamiltonian paths through the two copies of T_n of which T_{n+1} consists by definition, cf.* (28)*.*

Proof of Theorem 21 and Theorem 23. As before, we denote elements of $\underline{2}^{\underline{n}}$ as sequences such as 00101 (binary representation with most significant bit first) or, for clarity, by $0{\cdot}0{\cdot}1{\cdot}0{\cdot}1$. We use the endofunction neg on $\underline{2}^{\underline{n}}$, which simply replaces each 0 in a sequence by a 1 and vice versa; i.e. it sends the number i to $2^n - 1 - i$ (note that neg does not reverse the sequence, but the ordering on $\underline{2}^{\underline{n}}$).

[5] *Join* in the sense of the *join of categories* [19], which should not be confused with the join (coproduct) of objects in a preorder (cf. Definition 9).

Let us define endofunctions f_n and g_n on $\underline{2}^{\underline{n}}$, by induction on n. Note that, at this point, we do not talk about graph morphisms but only about functions between sets. The base cases of the induction are uniquely determined. We define f and g by

$$f_{n+1}(0 \cdot \vec{x}) :\equiv 0 \cdot f_n(\mathsf{neg}(\vec{x})) \qquad g_{n+1}(0 \cdot \vec{x}) :\equiv 0 \cdot \mathsf{neg}(g_n(\vec{x})) \tag{38}$$

$$f_{n+1}(1 \cdot \vec{x}) :\equiv 1 \cdot f_n(\vec{x}) \qquad g_{n+1}(1 \cdot \vec{x}) :\equiv 1 \cdot g_n(\vec{x}). \tag{39}$$

It is easy to calculate that, by induction, f and g are inverse to each other. We want to show that they extend to morphisms between preorders,

$$\widehat{f}_n : (\underline{2}^{\underline{n}}, <) \to T_n^* \qquad \widehat{g}_n : T_n^* \to (\underline{2}^{\underline{n}}, <). \tag{40}$$

To construct $\widehat{f}_n$ *and* the Hamiltonian path through the cube, it suffices to show: for $x, y : \underline{2}^{\underline{n}}$ with $x + 1 = y$, we have an edge $f_n(x) \to f_n(y)$.

We do induction on n. For $n = 0$, this is vacuously true (such x, y do not exist). For $n = n' + 1$, there are multiple cases:

- case $x = 0 \cdot x'$ and $y = 0 \cdot y'$: Then, the assumption gives us $x' + 1 = y'$ and we have to find an edge $0 \cdot f_n(\mathsf{neg}(x')) \to 0 \cdot f_n(\mathsf{neg}(y'))$. Looking at Definition 17, we can get this if we have $f_n(\mathsf{neg}(y')) \to f_n(\mathsf{neg}(x'))$. This holds by induction, since neg reverses the order which gives us $\mathsf{neg}(y') + 1 = \mathsf{neg}(x')$.
- case $x = 1 \cdot x'$ and $y = 1 \cdot y'$: Similar to the previous case, but nothing gets reversed.
- case $x = 0 \cdot x'$ and $y = 1 \cdot y'$: In this case, we have $x = 0111\ldots$ and $y = 1000\ldots$. We need to find an edge $0 \cdot f(\mathsf{neg}(111\ldots)) \to 1 \cdot f(000\ldots)$, which simplifies to $0 \cdot f(000\ldots) \to 1 \cdot f(000\ldots)$. This edge is directly given in (31).
- case $x = 1 \cdot x'$ and $y = 0 \cdot y'$: Contradicts with the assumption $x + 1 = y$.

This shows that there is a Hamiltonian path, and it is given by $\widehat{f}_n$. The definition of f as in (38,39) also shows that f_{n+1} consists of two copies of f_n, implying the last claim of Theorem 23. In order to prove Theorem 21, we need to construct $\widehat{g}_n$. It is enough to show that, for an edge from u to v in T_n, we have $g(u) \leqslant g(v)$. This follows by straightforward induction, going through the edges in Definition 17. But Theorem 21 implies that there is at most one Hamiltonian path. ◀

▶ Remark 24. Note that every vertex v in T_n is an endpoint of n non-trivial edges. The number of zeros in the binary representation in the "order number" of v (i.e. the value $g_n(v)$ in the proof of Theorem 21) equals the number of *outgoing* edges. Figure 7 shows this.

Analogously to Definition 8, we can now define the category of twisted graph morphisms:

▶ **Definition 25** (category $\rtimes_{\mathsf{grp}}$). *The category $\rtimes_{\mathsf{grp}}$ has natural numbers as objects, and morphisms from m to n are graph morphisms between twisted cubes:*

$$\mathsf{obj}(\rtimes_{\mathsf{grp}}) :\equiv \mathbb{N} \qquad \rtimes_{\mathsf{grp}}(m, n) :\equiv \mathsf{grp\text{-}hom}\,(T_m, T_n) \tag{41}$$

It is easy to see that the category $\rtimes_{\mathsf{grp}}$ has a version of connections. Since we are looking for a "twisted analogue" of $\square_{\mathsf{BCH}}^{\mathsf{op}}$, we need to refine it further. In Section 2, we have discussed the restriction to (meet and join)-preserving morphisms, and to dimension-preserving morphisms. It follows directly from Theorem 21 that every morphism in $\rtimes_{\mathsf{grp}}$ preserves all binary meets and joins, so this condition becomes trivial; it does not avoid connections. However, preserving dimensions is still a non-trivial condition which does avoid connections. The definition of equation (26) still works.

▶ **Definition 26** (category $\bowtie_{\mathsf{dim}}$). *The category $\bowtie_{\mathsf{dim}}$ has dimension-preserving maps between twisted cubes as morphisms:*

$$\mathsf{obj}(\bowtie_{\mathsf{dim}}) :\equiv \mathbb{N} \qquad \bowtie_{\mathsf{dim}}(m, n) :\equiv \Sigma(g : \mathsf{grp\text{-}hom}\,(T_m, T_n)).\mathsf{dim\text{-}pres}(g) \tag{42}$$

Note that the explanation of Remark 14 holds for the twisted cube category as well.

A consequence of Theorem 21 is that morphisms in $\bowtie_{\mathsf{dim}}$ cannot "swap dimensions". But an even stronger result holds, namely that surjective morphisms are unique:

▶ **Theorem 27.** *There is exactly one surjective morphism in $\bowtie_{\mathsf{dim}}(m, n)$ for $m \geqslant n$. (Clearly, there is none if $m < n$.)*

Proof. The key to the proof is Theorem 23. Clearly, the Hamiltonian path in T_m goes through all vertices. Due to surjectivity, its image has to go through all vertices of T_n. In other words, the T_m-Hamiltonian path has to be mapped to the T_n-Hamiltonian path. Since the graph morphisms that we consider preserve the dimension, the only edge in the T_m-path which can be mapped to the single edge in the first dimension in the T_n-path is just this single edge in the first dimension in the T_m-path; i.e. the middle edge has to be mapped to the middle edge. From here, it follows by induction that there can only be at most one surjective graph morphism.

What is left to show is that there actually is a surjective graph morphism if $m \geqslant n$. It is enough to construct a surjective graph morphism $f : \bowtie_{\mathsf{dim}}(n+1, n)$, from where we get any other by $(m-n)$-fold composition (0-fold composition is the identity). Such a graph morphism is given by

$$f(x_0 \dots x_{n-1} x_n) :\equiv (x_0 \dots x_{n-1}). \tag{43}$$

Since the directions of the edges do not depend on the very last dimension, this works (cf. Definition 19). ◀

An important consequence of the above result is that there is a unique way to degenerate a twisted cube. We do not go into the details here, but see the conclusions at the end of the paper. For now, we go into a different direction.

Let us write intv ("interval") for the finite set $\{0, 1, \star\}$. Of course, intv is isomorphic to $\underline{3}$, but referring to the last element as $\star$ helps the intuition, we hope.

▶ **Definition 28.** *A* face *of the twisted n-cube T_n is a function $f : \underline{n} \to \mathsf{intv}$. The* dimension *of a face, written $\mathsf{dim}(f)$, equals the number of times f takes $\star$ as value (i.e. the size of $f^{-1}(\star)$). The type of faces of dimension k is written as $\mathsf{faces}(n, k)$.*

The face $f : \underline{n} \to \mathsf{intv}$ represents the full subgraph of T_n of vertices on which f "matches" (a vertex $x_0 x_1 \dots x_{n-1}$ is matched if, for every i, we have $f(i) = x_i$ or $f(i) = \star$).

▶ **Lemma 29.** *The image of $f : \bowtie_{\mathsf{dim}}(m, n)$ is a face.*

Proof. This follows from the property of preserving the dimension as defined in (26). ◀

▶ **Lemma 30.** *The m-faces are the only injective maps $\bowtie_{\mathsf{dim}}(m, n)$:*

$$\mathsf{faces}(n, m) \simeq \Sigma(f : \bowtie_{\mathsf{dim}}(m, n)).\mathsf{is\text{-}inj}(f). \tag{44}$$

Proof. Every face gives rise to a canonical injective dimension-preserving morphism in the sense of Definition 13, as dictated by the inclusion of the full subgraph that the face represents into T_n. The fact that these are the only ones follows from Theorem 21 (we cannot "swap dimensions") and Lemma 29. ◀

As with Theorem 21 before, Lemma 30 is a result which is usually found in simplicial structures, but not in cubical ones. In any case, we now easily get:

▶ **Lemma 31** (factorisation of dimension preserving morphisms). *Given a morphism* $f : \bowtie_{\mathsf{dim}}(m, n)$*, there is exactly one way to write it as the composition* $f = \mathsf{inj}(f) \circ \mathsf{surj}(f)$ *of a surjective dimension preserving graph morphism followed by an injective one. This means that the map*

$$\big(\Sigma(k : \mathbb{N}). (\Sigma(h : \bowtie_{\mathsf{dim}}(k, n)).\mathsf{is\text{-}inj}(h)) \times (\Sigma(g : \bowtie_{\mathsf{dim}}(m, k)).\mathsf{is\text{-}surj}(g))\big) \to \bowtie_{\mathsf{dim}}(m, n) \quad (45)$$

$$(k, (h, i), (g, s)) \mapsto h \circ g \quad (46)$$

is an equivalence. Moreover, morphisms $\bowtie_{\mathsf{dim}}(m, n)$ *are in 1-to-1 correspondence with faces of* T_n *of dimension* $\leqslant m$*.*

Proof. A consequence of Lemma 29 is that the factorisation on the level of sets of vertices works. The second claim follows from the first: In (45), the k and the surjective map are uniquely determined (i.e. contractible components) by Theorem 27. By Lemma 30, injective maps correspond to faces. ◀

▶ Remark 32. It follows from Lemma 31 and the proof of Theorem 27 that all the non-empty fibres of a dimension-preserving morphism between twisted cubes have the same size. The reverse is the case as well: a morphism between twisted graphs where all non-empty fibres have the same size is dimension-preserving.

Another consequence of the above results is that $\bowtie_{\mathsf{dim}}$ can be given the structure of a *Reedy category* (cf. [14]). Recall that a Reedy category is a category R with a degree function $d : \mathsf{obj}(\bowtie_{\mathsf{dim}}) \to \mathbb{N}$ and two subcategories R^+ and R^-, such that:[6]

- both subcategories are *wide*, i.e. contain all the objects of R;
- every nonidentity morphism in R^+ raises the degree;
- every nonidentity morphism in R^- lowers the degree;
- and every morphism of R can be written as a morphisms in R^- followed by a morphism in R^+ in a unique way.

The reason why Reedy categories are interesting is that they enable certain inductive constructions. In the setting of type theory, they have been discussed by Shulman [26].

▶ **Theorem 33.** *The category* $\bowtie_{\mathsf{dim}}$ *is a Reedy category where the degree of an object is the object itself (recall that objects are natural numbers).* $\bowtie^+_{\mathsf{dim}}$ *is the subcategory of injective morphisms, and* $\bowtie^-_{\mathsf{dim}}$ *is the subcategory of surjective morphisms.*

Proof. The first three properties are clear, and the factorisation is given by Lemma 31. ◀

Finally, let us record an alternative representation of the category $\bowtie_{\mathsf{dim}}$ which does not go via graph morphisms.

▶ **Definition 34** (ternary notation: category $\bowtie_{\mathsf{tri}}$). *The category* $\bowtie_{\mathsf{tri}}$ *has natural numbers as objects, and a morphism from* m *to* n *is a function* $\underline{n} \to \mathsf{intv}$ *which takes* $\star$ *at most* m *times as image:*

$$\mathsf{obj}(\bowtie_{\mathsf{tri}}) :\equiv \mathbb{N} \qquad \bowtie_{\mathsf{tri}}(m, n) :\equiv \Sigma(f : \underline{n} \to \mathsf{intv}). f^{-1}(\star) \leqslant m \quad (47)$$

[6] Degrees can more generally be arbitrary ordinals, but $\mathbb{N}$ is sufficient in our case.

The identity morphisms are the functions that are constantly $\star$. To define the composition of $f : \bowtie_{\mathsf{tri}}(k, m)$ *and* $g : \bowtie_{\mathsf{tri}}(m, n)$*, we need to define a function* $g \circ f : \underline{n} \to \mathsf{intv}$ *(which is $\star$ at most k times). We define $(g \circ f)(i)$ by recursion on i, simultaneously with the values i' and b_i, as follows:*

$$(g \circ f)(i) :\equiv \begin{cases} g(i) & \text{if } g(i) \in \{0, 1\} \\ (f(i')) \;\mathsf{xor}\; b_i & \text{if } g(i) = \star \text{ and } f(i') \in \{0, 1\} \\ \star & \text{if } g(i) = \star \text{ and } f(i') = \star \end{cases} \tag{48}$$

where

- i' *is the number of occurrences of $\star$ in the sequence* $g(0), g(1), \dots, g(i-1)$*;*
- b_i *is* 1 *if the number of zeros in the sequence* $(g \circ f)(0), (g \circ f)(1), \dots, (g \circ f)(i-1)$ *is odd, and* 0 *if it is even.*

Note that a morphism in $\bowtie_{\mathsf{tri}}(m, n)$ can be represented as a sequence such as $01{\star}0{\star}10$ of length n which contains the symbol $\star$ at most m times, which is why we refer to it as *ternary notation*.

▶ Remark 35. There is a category of twisted semi-cubes, denoted by $\bowtie^+_{\mathsf{tri}}$, which is exactly the same as $\bowtie_{\mathsf{tri}}$ except that the number of $\star$ in the sequence must be exactly m, i.e. "$\leqslant$" is changed to "$=$" in the definition of $\bowtie_{\mathsf{tri}}(m, n)$. This category is equivalent to the subcategory of $\bowtie_{\mathsf{dim}}$, denoted as $\bowtie^+_{\mathsf{dim}}$, which consists of *injective* dimension-preserving graph homomorphisms. Note that this injectivity condition is equivalent to removing the reflexive edges from Definition 18.

If we remove the expression $(\mathsf{xor}\; b_i)$ in the definition of morphisms of $\bowtie^+_{\mathsf{tri}}$, then the category becomes equivalent to the category of standard cubes but without degeneracies and swapping dimensions. In other words, the expression $(\mathsf{xor}\; b_i)$ characterises "twisted-ness".

▶ **Theorem 36.** *The categories $\bowtie_{\mathsf{dim}}$, and $\bowtie_{\mathsf{tri}}$ are isomorphic, with the object part being the identity. In particular, we have:*

$$\bowtie_{\mathsf{dim}}(m, n) \simeq \bowtie_{\mathsf{tri}}(m, n) \tag{49}$$

Proof. As the following chain of equivalences:

$$\begin{aligned}
& & & \bowtie_{\mathsf{dim}}(m, n) \\
[\mathit{Lemma}\ 31] & & \simeq\ & \Sigma(k : \mathbb{N}).\, (\Sigma(h : \bowtie_{\mathsf{dim}}(k, n)).\mathsf{is\text{-}inj}(h)) \times (\Sigma(g : \bowtie_{\mathsf{dim}}(m, k)).\mathsf{is\text{-}surj}(g)) \\
[\mathit{Theorem}\ 27] & & \simeq\ & \Sigma(k : \mathbb{N}).\, (\Sigma(h : \bowtie_{\mathsf{dim}}(k, n)).\mathsf{is\text{-}inj}(h)) \times (k \leqslant m) \\
[\mathit{Lemma}\ 30] & & \simeq\ & \Sigma(k : \mathbb{N}).\, \mathsf{faces}(n, k) \times (k \leqslant m) \\
[\mathit{simplification}] & & \simeq\ & \Sigma(f : \underline{n} \to \mathsf{intv}).f^{-1}(\star) \leqslant m \\
& & \equiv\ & \bowtie_{\mathsf{tri}}(m, n)
\end{aligned}$$

When transported along this isomorphism, the composition of $\bowtie_{\mathsf{dim}}$ gets mapped to the composition of $\bowtie_{\mathsf{tri}}$, as required. ◀

4 Conclusions and Future Directions

We have suggested new representations of the BCH cube category and introduced a category of twisted cubes. It is natural to further study the similarities and differences between standard and twisted cube categories, and some new results will be presented in the upcoming PhD thesis of the first author.

As future work, we plan to examine algebraic descriptions via generators and relations. Such presentations exist for many different cube categories in the literature but, as far as we are aware, not for the BCH cube category. The closest suggestions available are the presentations by Antolini [5] and Newstead [20], which seem to be fairly easy to adapt to the BCH cube category. Interestingly, further adapting the generators to the *twisted* setting simplifies them significantly, which mirrors the fact that morphisms between twisted cubes cannot swap dimensions. Moreover, our Theorem 27 implies that degeneracies are unique: there is only one single way in which a twisted n-cube can be degenerated to get a twisted $(n+1)$-cube. A consequence is that we do not need to impose relations between different degeneracies.

This, we hope, will make it possible to develop the higher categorical structures that can be encoded as presheaves on the category of twisted cubes. An ultimate goal would be to model some form of *directed cubical type theory* mirroring the model by Bezem, Coquand, and Huber [8].

Another possible application of our twisted cube categories might be building a syntax for a parametric type theory or cubical type theory without an interval as suggested by Altenkirch and Kaposi [2]. A major difficulty in their development was the presence of multiple degeneracies, a problem which does not occur in the current work.

A further direction which may be worth exploring is to not consider set-valued presheaves, but type-valued presheaves instead. To facilitate this, we can consider the category of twisted semi-cubes mentioned in Remark 35. From there, type-valued presheaves can be encoded as Reedy-fibrant diagrams in a known style [27]. We can then add a condition reminiscent of Rezk's *Segal-condition* [24] by stating that the projection from twisted semi-cubical types to the sequence of types along the Hamiltonian path is an equivalence. This corresponds to saying that the partial n-cube with missing inner part and lid (cf. Figure 3) have a contractible type of fillers. It seems that this could be a first step towards the construction of composition and higher coherences, although further conditions seem to be necessary. The relation to the *(complete) semi-Segal types* by Capriotti and others [4, 10, 11] remains to be studied.

References

1. I. R. Aitchison. The geometry of oriented cubes. *ArXiv e-prints*, 2010. arXiv:1008.1714.
2. Thorsten Altenkirch and Ambrus Kaposi. Towards a Cubical Type Theory without an Interval. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:27, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TYPES.2015.3.
3. Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. Cartesian cubical type theory. URL: https://github.com/dlicata335/cart-cube/blob/master/cart-cube.pdf.
4. Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *ArXiv e-prints*, 2019. arXiv:1705.03307.
5. Rosa Antolini. Geometric realisations of cubical sets with connections, and classifying spaces of categories. *Applied Categorical Structures*, 2002.
6. Steve Awodey. A cubical model of homotopy type theory. *Annals of Pure and Applied Logic*, 2018.
7. Hans-Joachim Baues and Günther Wirsching. Cohomology of small categories. *Journal of pure and applied algebra*, 38(2-3):187–211, 1985. See also https://ncatlab.org/nlab/show/category+of+factorizations.
8. Marc Bezem, Thierry Coquand, and Simon Huber. A model of type theory in cubical sets. *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, 2014.

9 Ulrik Buchholtz and Edward Morehouse. Varieties of cubical sets. *Relational and Algebraic Methods in Computer Science*, 2017.

10 Paolo Capriotti. *Models of Type Theory with Strict Equality*. PhD thesis, School of Computer Science, University of Nottingham, Nottingham, UK, 2016. Available online at `arXiv:1702.04912`.

11 Paolo Capriotti and Nicolai Kraus. Univalent higher categories via complete semi-segal types. *Proceedings of the ACM on Programming Languages*, 2(POPL'18):44:1–44:29, December 2017. Full version available at `arXiv:1707.03693`. `doi:10.1145/3158132`.

12 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:34, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.TYPES.2015.5`.

13 Nicola Gambino and Christian Sattler. The frobenius condition, right properness, and uniform fibrations. *Journal of Pure and Applied Algebra*, 221(12):3027–3068, 2017.

14 Philip S Hirschhorn. *Model categories and their localizations*. American Mathematical Soc., 2009.

15 Simon Huber. *Cubical Interpretations of Type Theory*. PhD thesis, Department of Computer Science and Engineering, University of Gothenburg, 2016.

16 Chris Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of univalent foundations (after voevodsky). *ArXiv e-prints*, November 2012. To appear in the Journal of the European Mathematical Society. `arXiv:1211.2851`.

17 F William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. *Applications of Categorical Algebra*, 17:1–14, 1970. See also `https://ncatlab.org/nlab/show/twisted+arrow+category`.

18 Daniel R Licata and Robert Harper. 2-dimensional directed type theory. *Electronic Notes in Theoretical Computer Science*, 276:263–289, 2011.

19 Jacob Lurie. *Higher Topos Theory*. Annals of Mathematics Studies. Princeton University Press, Princeton, 2009. Also avaialabe online at `arXiv:math/0608040`; see also `https://ncatlab.org/nlab/show/join+of+categories`.

20 Clive Newstead. Cubical sets. URL: `math.cmu.edu/~cnewstea/notes/cubicalsets.pdf`.

21 Paige Randall North. Towards a directed homotopy type theory. *Electronic Notes in Theoretical Computer Science*, 347:223–239, 2019.

22 Andreas Nuyts. Towards a directed homotopy type theory based on 4 kinds of variance. Master's thesis, KU Leuven, 2015.

23 I. Orton and A. M. Pitts. Axioms for modelling cubical type theory in a topos. *Logical Methods in Computer Science*, 2018. Special issue for CSL 2016.

24 Charles Rezk. A model for the homotopy theory of homotopy theory. *Transactions of the American Mathematical Society*, 2001.

25 Emily Riehl and Michael Shulman. A type theory for synthetic ∞-categories. *Higher Structures*, 1(1), 2017. URL: `https://journals.mq.edu.au/index.php/higher_structures/article/view/36`.

26 Michael Shulman. The univalence axiom for elegant Reedy presheaves. *Homology, Homotopy and Applications*, 2015. `doi:10.4310/HHA.2015.v17.n2.a6`.

27 Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 2015.

28 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

29 Vladimir Voevodsky. Univalent foundations project. A modified version of an NSF grant application, 2010.