# Cartesian 🎲 category with involution

May 25, 2023

**Tesla Zhang**
teslaz@cmu.edu

**Abstract**

This note records what I learned in the recent week regarding cubical type theory, from multiple sources. In the end I included some guesses.

## Contents



## 1. Background

The definitions of various cube categories are done in lots of places [1, 2, 3] already, but those are too difficult for me – examples and motivations are probably given after cubical sets are defined, but I was already stuck in a very early stage. Carlo Angiuli's PhD thesis [4] has, instead, a very formal definition with graphical examples provided.

I find that the papers (reasonably) assuming familiarity with category of simplices, which is not the case for me. I did not fully understand the construction when I first saw it, because I cannot translate type theoretic intuition into this cube category. The thesis is much better, where I can find all the details, but I forgot the definitions a year after reading it.

Thankfully, David Berry explained me some constructions during the HoTT 2023 conference, including an extended version with involution, while we were trying to figure out the semantics of the cofibration theory of some variants of cubical type theory. This is a short note that records what he taught me (excluding the British accent).

I believe such a note will be valuable, because the first time I thought about a similar thing (see[1]) I failed to understand people's reaction and comments. I even asked some stupid follow-up questions on Discord, which made the conversation even more confusing. However, I feel like I am starting to understand them now, so why not serialize my learning process and share it with the world?

The construction starts from the idea of having a "cube" category (usually denoted $\square$, $\mathcal{C}$, or 🎲 if it's written by Favonia or Amélia) whose:

- Objects are unit cubes, which are roughly sets of vectors $\langle i, j, ...\rangle$ whose entries are within the unit interval $[0, 1]$, denoted $\mathbb{I}$. An $n$-cube has vectors of dimension $n$.
- Morphisms are special linear maps between these cubes, which is said to be "generated by degeneracies and face maps". This roughly means the full power of $\mathbb{R}$, like most arithmetic operations (addition, multiplication, etc.) are not allowed. Instead, we only map variables to either variables, 0,

[1] https://proofassistants.stackexchange.com/q/1906/32

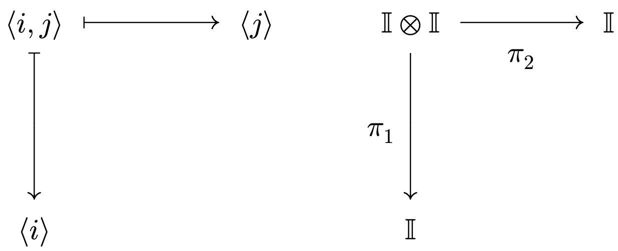
or 1. Depending on the variations of the cube category, there can be more operations. But these are the basics.

Examples of maps using additional operations:

$$\langle i, j\rangle \mapsto \langle i \wedge j\rangle$$
$$\langle i, j\rangle \mapsto \langle i, \neg j\rangle$$

Then, we may immediately see that only so little structures are needed that the notion of vectors is completely unnecessary. We instead work with an axiomatic imitator of $\mathbb{I}$. This is the conventional interval type with two endpoints.

This category is symmetric monoidal (moreover, it's semi-Cartesian as Trebor told me): the tensor product is roughly the Cartesian product of the interval. For instance, there can be diagrams like this:




Depending on the variations of the cube category, it may not support the full universal property of Cartesian product, but it is nevertheless symmetric monoidal. Furthermore, this is very similar to a Lawvere theory if our tensor product is actually Cartesian product (and the 0-dimensional cube is the terminal object).

### 1.1. Assumptions

This note assumes familiarity with the following concepts:

- Syntax of cofibrations and path types in cubical type theory,
- Operational semantics of Cartesian hcom,
- The idea of freely generating a algebra on a set,
- Categorical semantics of Martin-Löf type theory.

And the goal of this note is to discuss the semantics of cubical type theories at an elementary level.

Throughout the document, I'll be using $=$ for path-equality of terms and equality in cofibration theories, and $\equiv$ for judgmental equality of terms.

We overload the lambda syntax for path abstraction, similar to Cubical Agda [5]. The angle brackets $\langle - \rangle$ are reserved for vectors and introduction of product or sigma types.

The syntax of Cartesian hcom is like $\mathrm{hcom}^{i \rightsquigarrow j}(v)\{\varphi \mapsto u\}$, where $v$ is the base (the "floor"), $\varphi \mapsto u$ specifies the walls, $i \rightsquigarrow j$ gives the direction. The direction will be omitted if it's unimportant. In De Morgan cubical type theory, the direction is always $0 \rightsquigarrow 1$.

I appreciate Trebor and David Berry for pointing out several mistakes in the early versions of this note.

## 2. Motivation

Why study the semantics again, when the syntax of cubical type theory is already established?

The reason is that we were thinking about Cartesian cubical type theory with the involution operator from De Morgan cubical type theory. Then, should the following cofibration be considered false?

$$i = \neg i$$

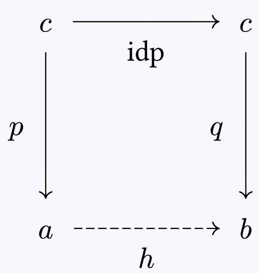
One may say: there is no closed term $i : \mathbb{I}$ that can make it true (unless we add a midpoint to $\mathbb{I}$, which has been investigated by Anders Mörtberg as some sort of ternary logic, but it did not solve the problems they were trying to solve, so it's no longer worked on), so it is quite false. One may also say: it sounds interesting because we can now express the notion of $\frac{1}{2}$ on $\mathbb{I}$.

**Example 2.1**: Consider the following composition problem (credits to David Berry):

$$p : a = c, q : b = c$$

$$h(i) := \mathrm{hcom}^{\neg i \rightsquigarrow i}(c) \begin{cases} \neg i = i \mapsto \lambda j.\, c \\ i = 0 \quad \mapsto p \\ i = 1 \quad \mapsto q \end{cases}$$

It is not hard to see that $h$ concatenates two paths $p$ and $q$. Diagrammatically,




From that it is quite obvious that $h : a = b$, which means

$$h(0) \equiv a$$
$$h(1) \equiv b$$

But now one may also extract the "midpoint" of $h$ by trapping it with a suitable context:

$$k = \neg k \vdash h(k) \equiv c$$

We were unsure how to explain this phenomenon, so it should be a good idea to discuss it from a semantic point of view.

# 3. Find a good definition

Following the ideas from free algebras, instead of vectors, we work with some *finite sets*, whose the elements are analogous to *variables*.

Then, to avoid dealing with renaming, let's think of these variables as some informal presentation of De Bruijn indices[2]. So the sets of identical cardinality are considered identical, and these sets will be denoted $[n]$ (due to David Berry), elements of $[n]$ include $0_n, 1_n, (n-1)_n$, like the indices of an array of length $n$ in the C programming language.

[2]If you're confused about this sentence (usually happens if you're not a type theorist), you may safely ignore this sentence.

**Clarification 3.1**: I used to think that $[n]$ are analogous to a finite product of some set $\mathbb{I}$ (i.e. $\mathbb{I}^n$), but this does not make any sense. First, in this perspective, the 1-dimensional cube $\mathbb{I}$ should be considered a singleton, which is already problematic: products of the singleton is still singleton, which trivializes everything.

However, we may think of $[n]$ as a finite coproduct of $[1]$, but we need to reverse the direction of the arrows.

It remains to define the maps. Using set-theoretic functions naïvely immediately leads to problems, like for a map $[1] \to [1]$, there is no way to map to the constant interval 0, because it does not belong to the codomain! To fix this, we define $\mathrm{Hom}_{🎲}(J, I)$ to be set-theoretic functions from $I$ to $J + \{0, 1\}$ (backwards!), where $+$ denotes disjointed union of sets.

In addition to the constants 0 and 1, there might be more operations, so we may replace $J + \{0, 1\}$ with a *free-algebra generated on* $J$ in the codomain.

This leads to the following definitions of some commonly seen categories of cubes from the literature. They are also defined in Carlo Angiuli's PhD thesis [4].

To make the notation less clumsy, we will write $\mathcal{C}$ for $\mathrm{Ob}(\mathcal{C})$ and $\mathcal{C}(A, B)$ for $\mathrm{Hom}_{\mathcal{C}}(A, B)$ when unambiguous.

**Definition 3.1**: The De Morgan cube category 🎲$_{\mathrm{DM}}$ consists of the following data:

- Objects are finite sets, denoted $[n]$, where $n \in \mathbb{N}$ is the cardinality.
- A hom-set 🎲$_{\mathrm{DM}}([m], [n])$ contains set-theoretic functions from $[n]$ to $\mathrm{DM}([m])$, where DM computes the free De Morgan algebra on a set.
- Composition. $g \circ f$ is the function that maps $x$ to $f(x)$ if $f(x) \in \{0, 1\}$, and otherwise we return $c(f(x))$ for:

$$c(x) = \begin{cases} g(i) & \text{if } x = i \text{ a variable} \\ c(u) \wedge c(v) & \text{if } x = u \wedge v \\ c(u) \vee c(v) & \text{if } x = u \vee v \\ \neg c(u) & \text{if } x = \neg u \end{cases}$$

In other words, we *monadically pipe* the functions to be composed, but when one of them returns 0 or 1, it will be taken as the output immediately.

The composition is called "Kleisli composition" and is obviously associative. $\mathrm{DM}(A)$ is written as $\mathsf{dM}(A)$ in [2, 3]. Another way to phrase the hom-sets (explained to me by Trebor) is to say that 🎲$_{\mathrm{DM}}(I, J)$ contains the De Morgan algebra homomorphisms from $\mathrm{DM}(I)$ to $\mathrm{DM}(J)$, and composition is inherited.

**Definition 3.2**: The Cartesian cube category $\square_C$ consists of the following data:

- Objects are finite sets.
- A hom-set $\square_C([m], [n])$ contains set-theoretic functions from $[n]$ to $[m] + \{0, 1\}$.
- Composition is again Kleisli composition as in $\square_{\mathrm{DM}}$.

Apparently, the Cartesian cube category has fewer structures, so the construction is simpler. Sophisticated cube maps in $\square_{\mathrm{DM}}$ are informally known as "De Morgan nonsense", credits to Amélia Liao. An example of such construction can be found in the Cubical Agda library[3], credits to Alexey Solovyev.

Even though with fewer structures in $\square_C$, the Cartesian cubical type theory has the "diagonal cofibration", which is very powerful and is lacking in $\square_{\mathrm{DM}}$.

# 4. New cube category

Now, we want a new cube category, which combines the two. The structures from $\square_{\mathrm{DM}}$ will be used in the cofibration theory, which has to be efficiently decidable for the sake of implementation.

I care about implementation!

## 4.1. Involution

The involution operator in $\square_{\mathrm{DM}}$ gives rise to involutive (that it *cancels out*) path reversal. In other words, it holds that

$$\left(p^{-1}\right)^{-1} \equiv p$$

with the following definition:

$$p^{-1} := \lambda i.\, p(\neg i)$$

This is the operator I want to add to the new cube category.

## 4.2. Connections

The connection operator in $\square_{\mathrm{DM}}$ may lead to cofibrations like $i = (i \land j)$, which is logically equivalent to $i \leq j$. The intuition behind is that $\land$ is similar to a "min" operator, while expanding $i = \min(i, j)$ into a case-analysis we get

$$i \equiv \begin{cases} i \text{ if } i \leq j \\ j \text{ if } i \geq j \end{cases}$$

It holds if and only if $i \leq j$.

I personally find these directed cubical cofibrations difficult to implement, so I want to avoid them for now. However, Reed Mullanix told me they are actually easy, and told me about `kado`[4]. So this may worth investigating in the future.

## 4.3. The new cube category

We start by the definition. The difference is only in the hom-sets.

[3]https://github.com/agda/cubical/blob/310a0956bb45ea49a5f0aede0e10245292ae41e0/Cubical/Data/Int/MoreInts/QuoInt/Properties.agda#L76

[4]https://github.com/RedPRL/kado

**Definition 4.3.1**: The Cartesian cube category with involutions 🎲$_{\mathrm{CI}}$, denoted 🎲 if no confusion, consists of the following data:

- Objects are finite sets, denoted $[n]$, where $n \in \mathbb{N}$ is the cardinality.
- A hom-set 🎲$_C([m], [n])$ contains set-theoretic functions from $[n]$ to $\{0, 1\} \times [m] + \{0, 1\}$, where $\langle 0, v\rangle$ stands for "mapping to $v$", and $\langle 1, v\rangle$ stands for "mapping to $\neg v$".
- Composition is again Kleisli composition as in 🎲$_{\mathrm{DM}}$:

$$(g \circ f)(x) \begin{cases} g(y) & \text{if } f(x) = \mathrm{inl}\langle 0, y\rangle \\ \overline{g}(y) & \text{if } f(x) = \mathrm{inl}\langle 1, y\rangle \\ \mathrm{inr}(1 - n) & \text{if } f(x) = \mathrm{inr}(n) \end{cases}$$

We may view $\{0, 1\} \times I$ as a "free $\neg$-algebra" generated on the set $I$. Suppose the left-introduction of $+$ is inl, and the right-introduction of $+$ is inr. Here are some examples of maps in 🎲.

**Example 4.3.1**: Interval constants are maps in 🎲$([0], [1]) := [1] \to \{0, 1\} \times [0] + \{0, 1\}$, which are just constant maps $\mathrm{inr}(0)$ and $\mathrm{inr}(1)$.

**Example 4.3.2**: The swap function is a map in 🎲$([1], [1])$, where

$$\begin{aligned} \mathrm{swap}(0) &:= \mathrm{inl}\langle 0, 1\rangle \\ \mathrm{swap}(1) &:= \mathrm{inl}\langle 0, 0\rangle \end{aligned}$$

**Example 4.3.3**: For a map $f :$ 🎲$([m], [n])$, its involution $\overline{f} :$ 🎲$([m], [n])$ is defined as

$$\overline{f}(x) = \begin{cases} \mathrm{inr}(1 - y) & \text{if } f(x) = \mathrm{inr}(y) \\ \mathrm{inl}\langle 1 - y, v\rangle & \text{if } f(x) = \mathrm{inl}\langle y, v\rangle \end{cases}$$

The $\overline{f}$ notation is due to David Berry, but it also looks like some Boolean algebra nonsense I've seen in an electrical engineering textbook.

Back to our type theory. Similar to [2], we define our version of cubical sets.

**Definition 4.3.2**: A *cubical set* is a presheaf on 🎲.

Then there is an obvious category of cubical sets Psh(🎲), whose morphisms are natural transformations. Amélia has a cool blog post[5] of cubical sets.

The category of cubical sets then become our category of contexts and substitutions, which admits dependent products, dependent sums, and a natural number object, see [2, Theorem 14]. The interval

[5]https://amelia.how/posts/cubical-sets.html

type (in the closed context) corresponds to the presheaf $I \mapsto \{0,1\} \times I + \{0,1\}$. Apart from those, it also has a subobject classifier (i.e. $\Omega(I)$ contains subfunctors of よ$(I)$), all small limits, etc., but those are not important for the purpose of this note.

### 4.4. Midpoint

David Berry proposed to really add a midpoint, and work in a setting where

$$\square(I, J) := I \to (\{0,1\} \times J + \{1, ½, 0\})$$

Note that the definition of composition scales to this case, but we may need to do something for the cofibration.

## 5. Nullable compositions

Before talking about the cofibration theory of our new cube category, we discuss an important optimization first. The content below is partially adapted from Evan Cavallo's explanation of some work in [6], including ghcom and validity.

There is a variation of hcom called ghcom, whose purpose is to reduce empty (aka "null") hcom terms, a major source of inefficiency in cubical type theory. They look like $\mathrm{hcom}(v)\{\}$.

Empty hcom terms can be generated by any partially filled (aka "nullable") hcom terms, because any such hcom terms can become empty by some substitution.

**Example 5.1**: The following hcom term is partially filled:

$$\mathrm{hcom}(v)\{i = 0 \mapsto u\}$$

Applying $[i := 1]$ will make it empty.

To get rid of partially filled hcom terms, we use the base to fill them. This requires the *negation* of cofibrations. We overload the notation of involutions $\neg$:

**Definition 5.1**: The *negation operator* $\neg$ on cofibrations is characterized by the following correspondence (credits to András Kovács and Jon Sterling):

$$\frac{\Gamma \vdash \neg\varphi}{\forall \Delta, (\sigma : \Delta \Rightarrow \Gamma), \Delta \vdash \varphi[\sigma] \Longrightarrow \Delta \vdash \bot}$$

In other words, the truthness of $\neg\varphi$ is equivalent to the situation that absurd contexts instantiate $\varphi$ to true.

The operational definition of $\neg$ is any function that satisfies the above correspondence. Apparently, it's different for different cube categories. Then, we use *double negation* to define when a composition is "filled":

**Definition 5.2**: A **valid** cofibration "has no missing faces". Formally speaking, $\varphi$ is valid if and only if $\neg\neg\varphi = \top$. We overload this terminology to also apply to hcom terms.

This is ported from [6, Definition 21] as well as [4, Definition 4.28].

**Theorem 5.1**: A valid hcom term will not become empty by any substitution.

**Theorem 5.2**: For cofibration $\varphi$, the negation satisfies *disjointedness*:

$$\neg\varphi \land \varphi = \bot$$
$$\neg\neg(\neg\varphi \lor \varphi) = \top$$

Then, we define a version of hcom, called "ghcom", that adds the base to missing faces.

**Definition 5.3**:

$$\mathrm{ghcom}(v)\{\varphi \mapsto u\} := \mathrm{hcom}(v)\begin{cases} \varphi & \mapsto u \\ \neg\varphi & \mapsto v \end{cases}$$

Suppose we have $\mathrm{hcom}(v)\{\varphi \mapsto u\}$, where $\varphi$ does not cover all faces. Then we simply replace hcom with ghcom, and it will never become empty by any substitution.

**Theorem 5.3**: A ghcom term is valid.

## 6. Questions

Back to the motivation of this note, we want to study the cofibration theory of 🎲. In particular, we want to answer the following questions about $i = \neg j$ and $i = \neg i$ (the latter is a special case of $i = \neg j$ under another cofibration $i = j$):

**Question 6.1**: What are the useful applications of the cofibration $i = \neg i$?

- The example of extracting a midpoint of a composition does not seem to solve/simplify any known problems in cubical type theory.
- Surely we can put this into a higher inductive type, but what are their semantics?
- I cannot think of any other interesting examples 😰

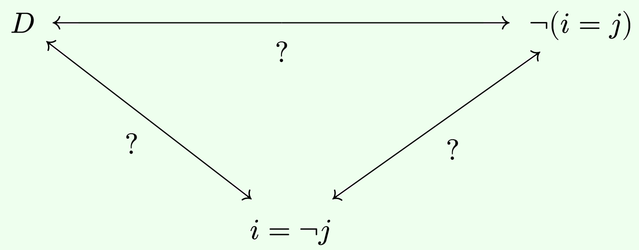
This is most likely open-ended.

**Question 6.2**: What happens if we set $i = \neg i$ to be just $\bot$?

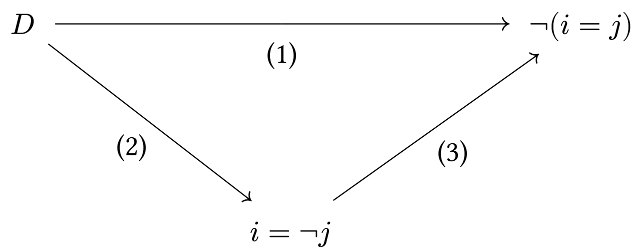
- If this implies some bad things, then we cannot have this rule.
- If this does not imply any bad things, then we can have this rule as a simplification.
- If it's implied by something else, then we probably want to add this rule as an optimization.

It is already clear that nothing implies $(i = \neg i) = \bot$, since there is a model of 🎲 using the real interval where it is not the case. If we have this rule, we refute this model, which is quite bad.

**Question 6.3**: Regarding ghcom, how are the following three concepts related, where $D$ is a shorthand of the formula $(i = 0 \land j = 1) \lor (i = 1 \land j = 0)$?




Using intuition from the real interval, the following implication relations are obviously admissible (without the presence of ½):




In the standard Cartesian cubical type theory, the inverse of $(1)$ is admissible, but I believe only the inverse of $(3)$ is admissible in 🎲 (which, if it's true, then we should be able to implement ghcom easily in this setting).

If there is a midpoint ½ : $\mathbb{I}$, then the cofibration theory gets more complicated, as $\neg(i = 0 \lor i = 1)$ may become $i =$ ½ (instead of $\bot$) and we may need to take care of it everywhere. So we have to reconsider all the compositions and we may need to replace them with ghcom (to fill the mid-face). It will also constitute a counterexample of $(3)$, where $i = j =$ ½ $\vdash i = \neg j$ but not $\neg(i = j)$.

It may help if we could answer the following question:

**Question 6.4**: For what substitution $\sigma$ does $(i = \neg j)\sigma$ equal to $\top$ and $\bot$?

**Question 6.5**: What to do with `Glue` if we have ½ : $\mathbb{I}$? Can we take the midpoint of an *equivalence*?

# Bibliography

[1] M. Bezem, T. Coquand, and S. Huber, “A model of type theory in cubical sets,” *19th Int. Conf. Types Proofs Programs (TYPES 2013)*, 2014. [Online]. Available: https://www.cse.chalmers.se/~coquand/mod1.pdf

[2] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg, “Cubical type theory: a constructive interpretation of the univalence axiom,” *Flap*, vol. 4, pp. 3127–3170, 2015.

[3] T. Coquand, S. Huber, and A. Mörtberg, “On higher inductive types in cubical type theory,” *Proc. 33rd Annu. ACM/IEEE Symp. Log. Comput. Sci. - LICS '18*, 2018, doi: 10.1145/3209108.3209197. [Online]. Available: https://dx.doi.org/10.1145/3209108.3209197

[4] C. Angiuli, “Computational semantics of cartesian cubical type theory,” Thesis, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2019.

[5] A. Vezzosi, A. Mörtberg, and A. Abel, “Cubical agda: a dependently typed programming language with univalence and higher inductive types,” *Proc. ACM Program. Lang.*, vol. 3, no. ICFP, Jul. 2019, doi: 10.1145/3341691.

[6] C. Angiuli, K.-B. H. (Favonia), and R. Harper, “Computational higher type theory III: univalent universes and exact equality,” 2017. [Online]. Available: http://arxiv.org/abs/1712.01800