Formalizing Higher Categories

Theofanis Chatzidiamantis Christoforidis

Born 21st December 2000 in Thessaloniki, Greece

August 8, 2024

Master's Thesis Mathematics

Advisor: Prof. Dr. Floris van Doorn

Second Advisor: Dr. Nima Rasekh

MATHEMATISCHES INSTITUT

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER

RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Contents

1. Introduction

Type theory is a foundational system for mathematics that replaces classical logic and set theory by a unified framework where inference rules and mathematical constructions exist on the same level. Let us elaborate on this: In type theory, everything is a type, including "objects", properties, implications, etc. Proving that a type has a certain property means providing a term, or a *witness*, of the type corresponding to that property.

The work presented in this thesis benefits from two big advantages of type theory against other formal systems:

- The constructive/computational style of type theory allows us to write formal proofs as computer programs in proof assistants; modulo errors in the implementation of the type theory itself, we can have a proof that is verifiably correct, in the full axiomatic sense.

- Different *type theories* can be developed as settings to work in different areas of mathematics. This is, in some sense, a question of *analytic* vs *synthetic* approaches to mathematics: The former of these is concerned with defining mathematical objects and deriving their desirable properties; The second, which we follow here, works by postulating that our objects have the desirable properties, and working from there. This thesis is concerned with one of these systems, *simplicial type theory* [RS17], a type theory made to reason about ∞-categories, and its dedicated proof assistant, Rzk.

So, why work with higher categories in the first place? Intuitively, higher categories are like generalized categories with objects, morphisms, and levels of n-morphisms between $(n-1)$-morphisms for each $n \geq 2$. Moreover, the usual properties of uniqueness and associativity of compositions are only satisfied up to weaker uniqueness conditions, which are witnessed by the morphisms one level above.

One category that we would like to generalize this way is the category of *topological spaces* $\mathcal{T}$op: A classic problem of $\mathcal{T}$op is that, while ordinary category theory works best when notions are considered up to isomorphisms, homotopy theory is concerned with studying spaces *up to homotopy equivalence* (or even up to weak homotopy equivalence!). We therefore want an analogous higher category where morphisms between morphisms correspond to homotopies, and we also have higher homotopies between homotopies, etc. All these levels of coherences are needed, as a lot of constructions in $\mathcal{T}$op are not homotopy invariant (see 2.3.3 for a quick example). Indeed, one can define the ∞-category of *spaces* or *anima* ($\mathcal{A}n$), where compositions are unique and associative only up to homotopy. Homotopy-coherent structures come up in many other variants: For example, when studying stable phenomena using the category of spectra, we can work with algebraic structures, which, once again, only satisfy their expected properties up to homotopy, thus leading us to the field of *higher algebra*.

We did not mention one thing: what are the "objects" in the category $\mathcal{A}n$? Topological spaces themselves have an internal higher homotopical structure, given by paths between points, homotopies between paths, etc. Note that, now, all paths are invertible. We can thus use the notion of an ∞-groupoid to capture this information about spaces. We can now go back and make a similar observation about ∞-categories: All higher morphisms, starting at homotopies, are equivalences. To remember this fact (and distinguish from more general notions), we often refer to ∞-groupoids as $(\infty, 0)$-categories and we say that the ∞-categories we work with are specifically $(\infty, 1)$-categories.

And why formalize ∞-categories synthetically? There is no a priori "correct" analytic definition of an ∞-category. Doing higher category theory usually involves working with concrete definitions usually referred to as *models* of ∞-categories, most often defined as objects in some category of *simplicial objects* that satisfy some coherence conditions. The two most used models are Boardman and Vogt's *quasicategories* [BV73] and Rezk's *complete Segal spaces* [Rez01]. These models, are, in some sense, homotopically equivalent, so it is a reasonable idea to try to present a general theory of ∞-categories, that has these constructions as its models. In set-based mathematics, such a theory is developed by Riehl and Verity in [RV22].

Even with the existence of such a generalization, a synthetic theory would still remove a lot of the technical work required to get any reasonable ∞-categorical machinery up and running (see, for example, Lurie's seminal work on quasicategories and higher topos theory [Lur09]). Simplicial type theory aims to reduce a lot of the basic study of higher categories to more familiar low-dimensional arguments, with the type theory keeping track of the higher coherences. Furthermore, the 2-categorical methods from [RV22] can be developed analogously in the type-theoretic context, as done, for example, by Buchholtz and Weinberger in [BW23].

From dependent type theory to homotopy type theory to simplicial type theory. The origin of all such homotopical thinking by type-theoretic means lies with Martin-Löf's *dependent type theory* [Mar71] and its *groupoid interpretation* by Hoffman and Streicher [HS98]. Among other things, Martin-Löf extended type theory with *intensional identity types*, i.e., type of witnesses to equalities between terms, which Hoffman and Streicher interpret as *paths*.

The next step was observing that these identity types can, just like paths and homotopies in topological spaces, be extended to capture higher homotopical structure, akin to the notion of ∞-groupoids discussed earlier. Together with the *univalence* axiom and the addition of *higher inductive types*, this type theory, now called *homotopy type theory* [Uni13], could now be interpreted in *spaces* [KL21], and later, in any ∞-topos [Shu19], i.e., any structure in which one can reasonably do homotopy theory.

The challenge now was moving from interpreting ∞-groupoids to $(\infty, 1)$-categories: adding the structure of spaces of non-invertible morphisms internal to a type, to create a so-called *directed type theory*. Types with directed morphisms could be defined in the setting of "book HoTT" already in [AKS15], but there was no way to capture the infinite levels of coherences internally, so category theory had to be done in terms of 1-categories. Riehl and Shulman manage to extend dependent type theory to a theory that can reason about ∞-categories by introducing a theory of *shapes*, out of which one can construct simplices which represent directed intervals, 2-cells witnessing compositions, etc. Thus, out of any type, we can take its hom-type, or the type of directed arrows, and by imposing conditions on it we can require that it is an ∞-category, with the standard semantics being in Reedy fibrant simplicial spaces, with ∞-categories corresponding to complete Segal spaces.

In this thesis, we present contributions to the Rzk formalization library of simplicial type theory. Note that not every type in simplicial type theory is an ∞-category, and in fact a lot of results can follow by imposing less strict assumptions. Finding the most general versions of results is another advantage of formalization of mathematics, where we want theorems that can be applied to as many places as possible, ideally proved only once. We also present some results that generalize for types with properties relating to higher-dimensional shapes. Here, we again take advantage of this axiomatic choice in simplicial type theory. A different approach, making all types ∞-categories, which should be more powerful but less general, is currently being developed by Cisinski, Nguyen, Walde and Cnossen in [Cis+].

Structure of the thesis

Chapter 2. We introduce our models for higher category theory and the tools we have to compare them using model categories. Before introducing quasicategories and complete Segal spaces, we also briefly discuss Kan complexes and their homotopy theory as motivation.

Chapter 3. We give a mostly formal presentation of homotopy type theory. Starting from the axiomatic system of dependent type theory,

Chapter 4. A short diversion, we discuss some 2-categorical examples that hint towards the relation of Rezk-completeness and univalence.

Chapter 5. After a brief presentation of the modifications one has to do to the Segal condition to get a reasonable type-theoretic definition, we introduce simplicial type theory and present some contributed formal proofs to the sHoTT library of formalizations in the Rzk proof assistant. Once again, we mainly focus the exposition on foundational aspects.

Chapter 6. Discussion regarding a new experimental project aimed at using simplicial type theory to formalize 2-Segal types, defined using generalized 3-horn filling conditions.

We assume familiarity with ordinary 1-category theory (Yoneda lemma, adjunctions, (co)limits). Knowledge of homotopy theory (homotopy groups, CW-complexes, Serre fibrations, truncations) is desirable for motivational purposes.

Acknowledgements

First and foremost, I want to express my deepest gratitude to my advisor Nima Rasekh. Nima suggested this project and taught me almost everything I know about higher categories. He is always there to discuss and answer any questions about this thesis and beyond, and was happy to share my projects and silly ideas with the simplicial type theory community. It has been a privilege working with you.

In the same vein, I would like to thank my supervisor Floris van Doorn, who was responsible for my first steps in the field of formal mathematics (albeit with a proof assistant that couldn't be more different), gave important pieces of feedback on my Rzk formalizations and my presentations, and organized a great seminar for all of us to learn about each other's work.

Moving to the broader research world, I have to mention some people who contributed invaluable mathematical insights to this project. In particular:

Thank you to Emily Riehl, for taking the time to review my formal proofs, giving feedback and providing me with a path towards formalizing dependent composition. All the ideas going into that proof were outlined by her.

Thank you to Jonathan Weinberger, for the great discussions explaining his work and formulating new project ideas. The two weeks he spent in Bonn were incredibly fruitful for both my understanding of the subject and thinking about any work on sHoTT I am currently doing and will do in the future.

Thank you to Nikolai Kudasov, for his technical advice on Rzk, for paying attention to my bug reports, and for his comments on my formal proofs and ideas.

Thank you to Fredrik Bakke, for his suggestions related to formalizing 2-Segal types.

This thesis has also been a product of some impeccable timing. At the very least, I have to draw attention to the fact that most people who could help me with this kind of mathematics spent some time in Bonn during the last couple of months (I guess that's also prior knowledge on Nima's part), the arrival of Floris van Doorn in the same month that I started my thesis, the existence of a seminar on higher Segal spaces during the previous semester.

The author is grateful to the Hausdorff Research Institute for Mathematics for the trimester program “Prospects of formal mathematics”.

I owe a lot to Tim, Daniël and Heiko forming a HoTT learning group during the last winter semester.

Finally, I want to thank everyone in the Bonn mathematics circles who thinks type theory is a bunch of incomprehensible nonsense. Nothing sharpens my understanding of (s)HoTT and constructive mathematics more than trying to explain myself to you.

2. Two Models for Higher Categories

2.1. A Quick Review of Model Categories

In this first chapter, our aim is to provide concrete "models" for the concepts of an ∞-groupoid and an ∞-category, and make comparisons: In the first case, we compare models of ∞-groupoids, one in Kan complexes and one in spaces, and in the second case, we compare models of ∞-categories to each other. One reason that both of these comparisons need to be made is to ensure that the objects we have defined provide a meaningful replacement for doing classical homotopy theory. Model categories, then, are the tools one uses to make such comparisons.

While the definition, which goes back to Quillen [Qui67] to generalize common results in algebraic topology and homological algebra, is quite long, the idea is relatively straightforward: When working homotopically we are concerned with notions of equivalence weaker than isomorphisms (In topology: weak homotopy equivalences, in algebra: chain homotopy equivalences, etc.), and model categories provide a coherent framework where one can use this weak equivalences as "normal" equivalences.

Along with weak equivalences, model categories come with two more classes of maps, representing the morphisms that interact with weak equivalences in desirable ways (lifting properties, etc.). Using these distinguished classes of maps, we then define homotopies and compare different model categories up to homotopy. As an imprecise slogan, we can say that "model categories present homotopy theories".

2.1.1 Definition. A ***model structure*** on a category $\mathcal{C}$ consists of three classes of morphisms called ***weak equivalences*** (denoted `W`), ***fibrations*** (denoted `Fib`), and ***cofibrations*** (denoted `Cof`), satisfying the following properties:

1. (*2-out-of-3 property* for weak equivalences). For morphisms $A \xrightarrow{f} B \xrightarrow{g} C$, if any two of $\{f, g, g \circ f\}$ are weak equivalences, so is the third.
2. `W`, `Fib` and `Cof` are *closed under retracts*: For any commutative diagram of the form

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "f"] \arrow[rr, bend left, "id_X"] & C \arrow[r] \arrow[d, "g"] & A \arrow[d, "f"] \\
B \arrow[r] \arrow[rr, bend right, "id_Y"'] & D \arrow[r] & B
\end{tikzcd}$$

(where we say *f is a retract of g*), if $g \in$ `W`/`Fib`/`Cof`, then $f \in$ `W`/`Fib`/`Cof` respectively.

3. For any commutative square

$$\begin{tikzcd} A \arrow[r] \arrow[d, "i"'] & B \arrow[d, "p"] \\ C \arrow[r] \arrow[ur, dotted, "\exists ?"] & D \end{tikzcd}$$

   if one of the following holds:

   - $i \in \mathtt{Cof}$ and $p \in \mathtt{Fib} \cap \mathtt{W}$, *or*
   - $i \in \mathtt{Cof} \cap \mathtt{W}$ and $p \in \mathtt{Fib}$,

   then there exists a lift making the diagram commute.

4. Every morphism $f : A \to B$ in C admits two factorizations:

   - $f = (A \xrightarrow{i} C \xrightarrow{p} B)$, where $i \in \mathtt{Cof}$ and $p \in \mathtt{Fib} \cap \mathtt{W}$.
   - $f = (A \xrightarrow{j} D \xrightarrow{q} B)$, where $j \in \mathtt{Cof} \cap \mathtt{W}$ and $q \in \mathtt{Fib}$.

A ***model category*** is a category with small limits and colimits equipped with a model structure.

2.1.2 Remark. In his original definition in [Qui67], Quillen only assumes that a model category has finite (co)limits. Small (co)limits are usually assumed in modern definitions, both to give access to more technical arguments and because the categories that usually concern us (spaces, simplicial objects, chain complexes) have small (co)limits.

2.1.3 Notation.

1. In particular, a model category $\mathcal{C}$ has an initial and a terminal object, which we denote by $\varnothing$ and $*$ respectively.
2. When working with diagrams, it is convenient to fix some notation for arrows in this section: We use ($\xrightarrow{\sim}$) for weak equivalences, ($\twoheadrightarrow$) for fibrations, and ($\hookrightarrow$) for cofibrations.

2.1.4 Definition. In a model category, morphisms in $\mathtt{Fib} \cap \mathtt{W}$ are called ***trivial fibrations*** (or acyclic fibrations) and morphisms in $\mathtt{Cof} \cap \mathtt{W}$ are called ***trivial cofibrations*** (or acyclic cofibrations).

2.1.5 Definition. Let $\mathcal{C}$ be a model category. An object A of C is called ***fibrant*** if the map $A \to *$ is a fibration. Dually, A is ***cofibrant*** if the map $\varnothing \to A$ is a cofibration. A is ***bifibrant*** if it is both fibrant and cofibrant.

2.1.6 Construction ((Co)fibrant replacement). Let $\mathcal{C}$ be a model category and A an object of $\mathcal{C}$. By [2.1.1, 4.], we can factor the map into $*$ as

$$\begin{tikzcd} A \arrow[rr] \arrow[dr, hook, "\sim"'] & & * \\ & RA \arrow[ur, two heads] & \end{tikzcd}$$

(note that RA is not necessarily unique). RA is now a fibrant object which is weakly equivalent to A. We call such an object RA together with a trivial cofibration $A \overset{\sim}{\hookrightarrow} RA$ a ***fibrant replacement*** of A.

Similarly, factoring
$$\begin{tikzcd}
\emptyset \arrow[rr] \arrow[dr, hook] & & A \\
& QA \arrow[ur, two heads, "\sim"'] &
\end{tikzcd}$$
results in a ***cofibrant replacement*** $QA \overset{\sim}{\twoheadrightarrow} A$.

2.1.7 Remark. We will also make a stronger assumption on (co)fibrant replacement: A lot of constructions may depend on a choice of factorization, so we assume that the constructions above assemble into *(co)fibrant replacement functors* $R, Q : \mathcal{C} \to \mathcal{C}$. This is reasonable to assume as, for example, functorial factorizations like this exist for

- All combinatorial model categories.
- Classes produced by a *small object argument* (e.g. [Hir03, 10.5.16]).

2.1.8 Lemma. *Any class of morphisms defined via a right lifting property (in particular,* $\mathtt{Fib}$ *and* $\mathtt{Fib} \cap \mathtt{W}$ *in a model category) is stable under pullback.*

Proof. Assume we have a pullback diagram

$$\begin{tikzcd}
A \times_C B \arrow[r, "k"] \arrow[d, "r^* f"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\
B \arrow[r, "r"'] & C
\end{tikzcd}$$

and f has the right lifting property against a morphism $g : X \to Y$. Consider the diagram

$$\begin{tikzcd}
X \arrow[r, "l"] \arrow[d, "g"] & A \times_C B \arrow[r, "k"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\
Y \arrow[r, "s"'] \arrow[urr, dotted, "\exists h"] & B \arrow[r, "r"'] & C
\end{tikzcd}$$

where, by assumption, we have a lift $h : Y \to A$ with $r \circ s = f \circ h$ and $k \circ l = h \circ g$. By the universal property of $A \times_C B$, we get a unique map $\tilde{h} : Y \to A \times_C B$ making everything on the right commute. Then $\tilde{h}$ also commutes with the maps on the left by the uniqueness part of the universal property of the pullback. □

2.1.9 Proposition. *Let* $\mathcal{C}$ *be a model category. Then* $\mathtt{Cof}$ *consists precisely of the morphisms with the left lifting property against* $\mathtt{Fib} \cap \mathtt{W}$. *Dually,* $\mathtt{Fib}$ *consists precisely of the morphisms with the right lifting property against* $\mathtt{Cof} \cap \mathtt{W}$.

Proof. See e.g. [Hir03, Proposition 7.2.3]. □

2.1.10 Corollary. $\mathtt{Fib}$ *and* $\mathtt{Cof}$ *are closed under composition.*

Using (2.1.9) and with a little more work, we can show that model structures can be uniquely determined by various distinguished classes of objects and morphisms. We will take advantage of these facts later, using the simplest classes to determine the rest of the structure, making the exposition easier to follow. For a detailed discussion, see [Joy, Appendix E].

2.1.11 Proposition. *A model structure on a category $\mathcal{C}$ is uniquely determined by any of the following:*

1. *Any two of the classes* `W`, `Fib`, `Cof`.
2. *Cofibrations and fibrant objects (dually, fibrations and cofibrant objects).*
3. *Trivial fibrations and fibrant objects (dually, trivial cofibrations and cofibrant objects).*

Proof sketch. The key step is showing that `Fib` and `Cof` determine `W`. We can get $\mathtt{Fib} \cap \mathtt{W}$ and $\mathtt{Cof} \cap \mathtt{W}$ by (2.1.9). By the 2-out-of-3 property for weak equivalences, we see that a morphism in $\mathcal{C}$ is a weak equivalence if it factors as a trivial cofibration followed by a trivial fibration (Just use a factorization guaranteed by the axioms and 2-out-of-3), thus we have determined `W`. □

2.1.12 Example. Every (co)complete category $\mathcal{C}$ can be made into a model category via the *trivial model structure*, where the weak equivalences and every morphism is both a fibration and a cofibration.

2.1.13 Example (Model structures on $\mathcal{T}$op). A category may be equipped with more than one nontrivial model structure. There are two model structures on the category of topological spaces, using two commonly studied classes of equivalences and fibrations:

1. The ***classical/Quillen model structure*** on $\mathcal{T}$op, where
    - `W` = {weak equivalences}
    - `Fib` = {Serre fibrations}
    - `Cof` = {retracts of relative cell complexes} (in particular, all CW-inclusions)
2. The ***Strøm model structure*** on $\mathcal{T}$op, where
    - `W` = {homotopy equivalences}
    - `Fib` = {Hurewicz fibrations}

2.1.14 Definition. Let $\mathcal{C}$ be a model category and D a small category. Define "model structures" on the functor category Fun($\mathcal{D}$, $\mathcal{C}$) with the following classes of maps:

1.
    - `W` = {objectwise weak equivalences}
    - `Cof` = {objectwise cofibrations}

    which, if it defines a model structure, we call the ***injective model structure***.
2.
    - `W` = {objectwise weak equivalences}

- `Fib` = {objectwise fibrations}

which, if it defines a model structure, we call the ***projective model structure***.

2.1.15 Remark. One advantage of working with the projective and injective model structures is that they are very simple to define. However, one needs to check that they exist. Once condition that guarantees their existence is $\mathcal{C}$ being a *combinatorial* model category, which roughly means that it is generated by cofibrations using "small data" (For details, see [Lur09, Section A.2.6]). The "base" model categories we will be concerned with are combinatorial, so we may assume that these model structures exist.

The homotopy category of a model category

We introduced model categories in order to have a consistent method of working with objects up to weaker forms of equivalences. The homotopy category of a model category realizes this, using the distinguished classes of morphisms to define the appropriate notion of homotopy between two maps, and then working with morphisms modulo homotopy relations. Moreover, the homotopy category is equivalent to the category one gets from formally inverting the weak equivalences (localizations can have size issues; in this case it is a category!), justifying the fact that weak equivalences are now our "proper" equivalences.

2.1.16 Definition. The ***homotopy category*** of a model category $\mathcal{C}$ is the category $\mathcal{C}[\mathtt{W}^{-1}]$ obtained by formally inverting the weak equivalences.

For an arbitrary collection of maps in a category $\mathcal{C}$, this definition has set-theoretic issues: Localizations $\mathcal{C}[M^{-1}]$ may not be presented in a way that guarantees the hom-classes in $\mathcal{C}[M^{-1}]$ to be sets. However, when inverting weak equivalences in a model category, we can provide an explicit construction of a category that realizes this definition.

Constructing the homotopy category. We have to say what two maps being homotopic means in this context. A definition of homotopy first requires a path, so this is where we begin.

2.1.17 Definition. Let $\mathcal{C}$ be a model category and X and object of $\mathcal{C}$.

- A ***path object*** of X is a factorization of the diagonal

$$\begin{tikzcd} X \arrow[r, "\sim", "i"'] \arrow[rr, bend left, "\Delta_X"] & P(X) \arrow[r, two heads, "p"'] & X \times X \end{tikzcd}$$

- Dually, a ***cylinder object*** of X is a factorization of the codiagonal

$$\begin{tikzcd} X \coprod X \arrow[r, hook, "j"'] \arrow[rr, bend left, "\nabla_X"] & C(X) \arrow[r, "\sim", "q"'] & X \end{tikzcd}$$

As with (co)fibrant replacement, these factorizations exist by definition.

2.1.18 Definition. Let $\mathcal{C}$ be a model category and $f, g : X \to Y$ morphisms in $\mathcal{C}$. We say

- f is ***right homotopic*** to g ($f \simeq_R g$) if there is a commutative diagram

$$\begin{tikzcd} X \arrow[r] \arrow[dr, "{(f,g)}"'] & P(Y) \arrow[d, two heads] \\ & Y \times Y \end{tikzcd}$$

  where $P(Y) \twoheadrightarrow Y \times Y$ is a path object.

- f is ***left homotopic*** to g ($f \simeq_L g$) if there is a commutative diagram

$$\begin{tikzcd} X \coprod X \arrow[r, hook] \arrow[dr, "{(f,g)}"'] & C(X) \arrow[d] \\ & Y \end{tikzcd}$$

  where $X \coprod X \hookrightarrow C(X)$ is a cylinder object.

2.1.19 Proposition. *Let $\mathcal{C}$ be a model category $f, g : X \to Y$ morphisms in $\mathcal{C}$. Then:*

1. *If X is cofibrant, $\simeq_L$ is an equivalence relations on* $\mathrm{Hom}_{\mathcal{C}}(x, y)$*.*
2. *If Y is fibrant, $\simeq_R$ is an equivalence relations on* $\mathrm{Hom}_{\mathcal{C}}(x, y)$*.*
3. *If X is cofibrant and Y is fibrant, then $f \simeq_L g$ if and only if $f \simeq_R g$.*
4. *Homotopy respects compositions.*

Proof. See e.g. [Hir03, 7.4.5, 7.4.9]. ☐

2.1.20 Remark. Under the same assumptions, one can also prove that homotopy relations are independent of the choice of path/cylinder object.

2.1.21 Definition. For a model category $\mathcal{C}$, define the category $\mathrm{Ho}(\mathcal{C})$ with

- Objects: the objects of $\mathcal{C}$.
- $\mathrm{Hom}_{\mathrm{Ho}(\mathcal{C})}(X, Y) := \mathrm{Hom}\big(Q(R(X)), Q(R(Y))\big)\big/\sim$

As in spaces, we say that a morphism is a *homotopy equivalence* if it has an inverse in the homotopy category.

2.1.22 Theorem (Model category Whitehead). *In a model category, every weak equivalence between bifibrant objects is a homotopy equivalence.*

Proof sketch. As in the proof of (2.1.11), a weak equivalence $f : X \to Y$ is a composite $X \xhookrightarrow[i]{\sim} Z \overset{\sim}{\underset{p}{\twoheadrightarrow}} Y$ so it suffices to prove that trivial (co)fibrations have homotopy inverses. Then we can solve lifting problems of the form

$$\begin{tikzcd}
\varnothing \arrow[r] \arrow[d] & A \arrow[d, two heads, "\sim"] \\
B \arrow[r, "id_B"'] \arrow[ur, dotted] & B
\end{tikzcd}$$

and its dual for trivial cofibrations. □

Thus we have constructed a category $\mathrm{Ho}(\mathcal{C})$ that is equivalent to $\mathcal{C}[\mathtt{W}^{-1}]$. Note that we could also use the category of bifibrant objects in $\mathcal{C}$ together with homotopy classes of morphisms, and we would still get an equivalence by Whitehead and the fact that $X \to Q(R(X))$ is a weak equivalence.

2.1.23 Example (Some algebra, [Qui67], [DS95]). Let R be a ring. We can define a model structure on the category of bounded chain complexes $\mathrm{Ch}_{\geq 0}(R)$ such that:

- The weak equivalences are the quasi-isomorphisms.
- The fibrations are the levelwise epimorphisms.
- The cofibrations are the levelwise monomorphisms with a projective cokernel.

Thus, repeating the homotopy category construction we obtain a category that formally inverts the quasi-isomorphisms. In other words, we have defined the derived category $D(R)$!

We now define the structure-preserving functors that we use to compare homotopy theories presented by model categories. We mainly follow the exposition of [Lur09, Appendix A].

2.1.24 Proposition. *Let* $\begin{tikzcd} \mathcal{C} \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, shift left, "G"] \end{tikzcd}$ *be an adjoint pair between two model categories* $\mathcal{C}, \mathcal{D}$. *The following are equivalent:*

1. *F preserves cofibrations and trivial cofibrations.*
2. *G preserves fibrations and trivial fibrations.*
3. *F preserves cofibrations and G preserves fibrations.*
4. *F preserves trivial cofibrations and G preserves trivial fibrations.*

Proof. See e.g. [Hir03, Proposition 8.5.3]. □

2.1.25 Definition. An adjunction $(F \dashv G)$ between model categories is a ***Quillen adjunction*** if any of the equivalent conditions in (2.1.24) are satisfied.

2.1.26 Lemma. *Let $(F \dashv G)$ be a Quillen adjunction. Then:*

1. *F takes weak equivalences between cofibrant objects to weak equivalences.*
2. *G takes weak equivalences between fibrant objects to weak equivalences.*

Then, if we have (co)fibrant replacement functors, we can compose with them to get induced functor

2.1.27 Definition. Let $(F \dashv G)$ be a Quillen adjunction. We define

1. The (total) ***left derived functor*** $\mathbb{L}F : \mathrm{Ho}(\mathcal{C}) \to \mathrm{Ho}(\mathcal{D})$,

$$\mathbb{L}F \coloneqq \mathrm{Ho}(\mathcal{C}) \xrightarrow{\mathrm{Ho}(Q)} \mathrm{Ho}(\mathcal{C}) \xrightarrow{\mathrm{Ho}(F)} \mathrm{Ho}(\mathcal{D})$$

2. The (total) ***right derived functor*** $\mathbb{R}G : \mathrm{Ho}(\mathcal{D}) \to \mathrm{Ho}(\mathcal{C})$,

$$\mathbb{R}G \coloneqq \mathrm{Ho}(\mathcal{D}) \xrightarrow{\mathrm{Ho}(R)} \mathrm{Ho}(\mathcal{D}) \xrightarrow{\mathrm{Ho}(G)} \mathrm{Ho}(\mathcal{C})$$

2.1.28 Remark. For functors $F : \mathcal{C} \longleftrightarrow \mathcal{D} : G$ between model categories, we can define derived functors by a characterizing universal property: In particular, we can define $\mathbf{L}F : \mathrm{Ho}(\mathcal{C}) \to \mathcal{D}$ as the *right Kan extension* of F along $\mathcal{C} \to \mathrm{Ho}(\mathcal{C})$ and $\mathbf{R}G$ as the *left Kan extension* of G along $\mathcal{D} \to \mathrm{Ho}(\mathcal{D})$ (which may not exist for arbitrary F, G). Then compose with the projections to the homotopy categories to get total derived functors as above.

2.1.29 Proposition. *Let $(F \dashv G)$ be a Quillen adjunction. Then the derived functors $\mathbb{L}F : \mathrm{Ho}(\mathcal{C}) \to \mathrm{Ho}(\mathcal{D})$ and $\mathbb{R}G : \mathrm{Ho}(\mathcal{D}) \to \mathrm{Ho}(\mathcal{C})$ form an ajoint pair*

$$\begin{tikzcd}
\mathrm{Ho}(\mathcal{C}) \arrow[r, bend left, "\mathbb{L}F"] \arrow[r, phantom, "\perp"] & \mathrm{Ho}(\mathcal{D}) \arrow[l, bend left, "\mathbb{R}G"]
\end{tikzcd}$$

2.1.30 Proposition. *Let* $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$ *be a Quillen adjunction between two model categories $\mathcal{C}, \mathcal{D}$. The following are equivalent:*

1. *The left derived functor $\mathbb{L}F : \mathrm{Ho}(\mathcal{C}) \to \mathrm{Ho}(\mathcal{D})$ is an equivalence of categories.*
2. *The right derived functor $\mathbb{R}G : \mathrm{Ho}(\mathcal{D}) \to \mathrm{Ho}(\mathcal{C})$ is an equivalence of categories.*
3. *For every cofibrant object $C \in \mathcal{C}$ and every fibrant object $D \in \mathcal{D}$, a morphism $C \to \mathbb{R}D$ is a weak equivalence in $\mathcal{C}$ precisely when the adjunct morphism $\mathbb{L}C \to D$ is a weak equivalence in $\mathcal{D}$.*
4. *The unit and counit of the induced adjunction $(\mathbb{L}F \dashv \mathbb{R}G)$ are weak equivalences.*

Proof. See e.g. [Lur09, Proposition A.2.5.1]. ☐

2.1.31 Definition. A Quillen adjunction $(F \dashv G)$ between model categories is a ***Quillen equivalence*** if any of the equivalent conditions in (2.1.30) are satisfied.

2.2. Simplicial Objects and Homotopy Theory

A Reminder on Simplicial Objects

We begin by defining the objects that will concern us for the rest of this chapter. It is not immediately clear why this is the way to go when trying to define structures that adequately replace classical homotopy theory; We will spend the next few sections to motivate this correspondence and make some of it precise.

2.2.1 Definition.

1. We denote by Δ the ***simplex category*** with objects $[n] := \{0, \dots, n\}$ for all $n \in \mathbb{N}$ and morphisms all non-decreasing maps.

2. Let $\mathcal{C}$ be a category. The category of ***simplicial objects*** in $\mathcal{C}$ is the functor category $\mathrm{Fun}(\Delta^{op}, \mathcal{C})$.

We will mainly be concerned with two special cases: The category $s\mathrm{Set} := \mathrm{Fun}(\Delta^{op}, \mathrm{Set})$ of ***simplicial sets*** and the category of ***simplicial spaces***

$$s\mathcal{S} := \mathrm{Fun}(\Delta^{op}, s\mathrm{Set})$$

We identify $s\mathcal{S}$ with the category of ***bisimplicial sets***, i.e., the functor category

$$\mathrm{Fun}(\Delta^{op} \times \Delta^{op}, \mathrm{Set})$$

For a simplicial space X and $n \in \mathbb{N}$, the simplicial set X_n itself consists of sets $X_{nm} := X_n(m)$.

2.2.2 Definition. In Δ, we define the ***coface*** $\delta_n^i : [n-1] \to [n]$ to be the unique strictly monotone map that skips i, and the ***codegeneracy*** $\sigma_n^j : [n+1] \to [n]$ to be the map that repeats j in positions j and $j+1$.

We can form their induced maps in any simplicial object X:
We get ***face maps***

$$d_i^n := (\delta_n^i)^* : X_n \to X_{n-1}$$

and ***degeneracy maps***

$$s_j^n := (\sigma_n^j)^* : X_n \to X_{n+1}$$

We usually skip the n for convenience whenever it is implied.

2.2.3 Remark. It is easy to check that $d_i^{n+1} \circ s_i^n = id_{X_n}$ for any simplcial object X. Geometrically, we will see that we interpret face maps as projections to a face and degeneracies as creating "trivial" higher-dimensional edges.

We now provide some combinatorial computations to give a more concrete description of a simplicial object.

2.2.4 Proposition (cosimplicial identities).

$$\begin{aligned}
\delta^j \circ \delta^i &= \delta^i \circ \delta^{j-1} && \textit{if } i < j \\
\sigma^j \circ \sigma^i &= \sigma^i \circ \sigma^{j+1} && \textit{if } i \leq j \\
\sigma^j \circ \delta^j &= \sigma^j \circ \delta^{j+1} = id && \\
\sigma^j \circ \delta^i &= \delta^{i-1} \circ \sigma^j && \textit{if } i > j+1 \\
\sigma^j \circ \sigma^i &= \sigma^i \circ \sigma^{j+1} && \textit{if } i \leq j
\end{aligned}$$

Then, in any simplicial object, the d_i and s_i satisfy the duals of these identities, which we call the ***simplicial identities*** (See [GJ99, 1.2, 1.3]).

2.2.5 Proposition. *Every morphism in Δ can be uniquely written as a composition of cofaces and codegeneracies.*

Proof. See e.g. [Mac98, Section II.5]. ☐

Let us now take a closer look at simplicial objects: A functor $X : \Delta^{op} \to \mathcal{C}$ consists of an object $X_n := X(n)$ for every $n \in \mathbb{N}$, and, on morphisms, (2.2.5) tells us that it is enough to specify the behavior of X on the coface and codegeneracies, i.e., its face and degeneracy maps.

2.2.6 Remark. Using (2.2.5), it is easy to show that the definition of a simplicial set as a functor is, in fact, equivalent to specifying such a sequence of objects and morphisms satisfying the simplicial identities. In particular, Δ itself is completely determined by the cosimplicial identities.

2.2.7 Example (Nerve of a category). Let $\mathcal{C}$ be a category. Consider the posets $[n]$, $n \in \mathbb{N}$, as categories $(0 \to 1 \to \cdots \to n)$. Define the ***nerve*** $N\mathcal{C}$ of $\mathcal{C}$ to be the simplicial set by setting

$$N\mathcal{C}_k := \mathrm{Fun}([k], \mathcal{C})$$

(meaning that $N\mathcal{C}_k$ consists of *tuples of composable morphisms* $A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{k-1}} A_k$) and letting the face d_i^n be the map removing A_n and composing the remaining maps if $n \neq 0, k$, and the degeneracy s_i^n be the map duplicating A_n with an identity map in between.

2.2.8 Remark. One nice property of functor categories $\mathrm{Fun}(\mathcal{D}, \mathcal{C})$ is that (co)limits exist and can be computed pointwise if they exist in $\mathcal{C}$. Thus $s\mathrm{Set}$ and $s\mathcal{S}$ are (co)complete.

We now introduce some notation for the representable functors in $s\mathrm{Set}$.

2.2.9 Definition. Let $n \in \mathbb{N}$. The ***standard*** n***-simplex*** Δ^n is the functor $\mathrm{Hom}(-, [n])$.

Then, an immediate consequence of the Yoneda lemma is that there is a natural bijection

$$\mathrm{Hom}_{s\mathrm{Set}}(\Delta^n, X) \cong X_n$$

for every simplicial set X. We call a map $\Delta^n \to X$ (or, equivalently, an element of X_n) an *n-simplex* of X.

2.2.10 Example. Note that there is only one map $[n] \to [0]$ for every n, thus Δ^0 is a final object of sSet. Knowing this, it makes sense to think of the 0-simplices of a simplicial sets as its points.

To what extent can we repeat this for simplicial spaces? There are two ways that we can think of a simplicial set X as a simplicial space:

- The ***vertical embedding*** $i_F(X)_{nm} := X_n$, and
- The ***horizontal embedding*** $i_\Delta(X)_{nm} := X_m$

2.2.11 Definition. For $n \in \mathbb{N}$ we define $F(n) := i_F(\Delta^n)$ and $\Delta[n] := i_\Delta(\Delta^n)$.

We now recall an important fact about presheaf categories.

2.2.12 Proposition. *Let $\mathcal{D}$ be a small category. Then the category* $\mathrm{Fun}(\mathcal{D}^{op}, \mathcal{S}\mathrm{et})$ *is cartesian closed, with exponential given by the formula*

$$B^A(x) = \mathrm{Nat}(\mathrm{Hom}_{\mathcal{D}}(-, x) \times A, B) \tag{*}$$

for any functors $A, B : \mathcal{D}^{op} \to \mathcal{S}\mathrm{et}$ and $x \in \mathrm{Obj}(\mathcal{D})$.

Proof. See, for example, [MM92, I.6, Proposition 1] ☐

Let us first apply this to sSet: (*) tells us precisely that

$$(Y^X)_n = \mathrm{Hom}_{s\mathrm{Set}}(\Delta^n \times X, Y)$$

For $s\mathcal{S}$, things get a little more complicated: If Z and W are simplicial spaces, we first have to look at them as bisimplicial sets to have a category where (*) applies. The exponential turns out to be

$$(W^Z)_{nm} = \mathrm{Hom}_{s\mathcal{S}}(F(n) \times \Delta[m] \times Z, W)$$

We define its 0-th level to be the ***mapping space***

$$\mathrm{Map}_{s\mathcal{S}}(Z, W) = (W^Z)_0 \cong \mathrm{Hom}_{s\mathcal{S}}(\Delta[\bullet] \times Z, W)$$

Note that this is a simplicial set.

We can now reapply the Yoneda lemma to this mapping space and get an isomorphism of simplicial sets

$$\mathrm{Map}_{s\mathcal{S}}(F(n), W) \cong W_n$$

Topology and Some Special Simplicial Sets

Having all the necessary notation worked out, it is now time to give a geometric interpretation of all the notions we introduced and work out how simplicial sets relate to homotopy theory. First, we will define a way to go from topological spaces to simplicial sets and back again.

2.2.13 Definition. Let $n \in \mathbb{N}$. The ***topological n-simplex*** is the space

$$|\Delta^n| := \{(t_0, t_1, \dots, t_n) \in [0,1]^{n+1} \, : \, t_0 + t_1 + \cdots + t_n = 1\}$$

with the subspace topology inherited from $\mathbb{R}^{n+1}$.

$|\Delta^n|$ is a contractible space, the boundary of which has $n+1$ vertices.
We can define a covariant functor $\Delta \to \mathcal{T}\mathrm{op}$ in the following way:

- On objects, $[n] \mapsto |\Delta^n|$.
- On morphisms, $f \, : \, [n] \to [m]$ is sent to $f_* \, : \, |\Delta^n| \to |\Delta^m|$ defined by

  $$f_*(t_0, \dots t_n) = (\sum_{j \in f^{-1}(0)} t_j, \dots , \sum_{j \in f^{-1}(n)} t_j)$$

  where sums over empty sets are taken to be 0.

To go from $\mathcal{T}\mathrm{op}$ to *s*Set, we have a familiar construction from algebraic topology:

2.2.14 Definition. Let T be a topological space. Its ***singular simplicial set*** $\mathrm{Sing}_\bullet(T)$ is defined by

- $\mathrm{Sing}_n(T) := \mathrm{Hom}_{\mathcal{T}\mathrm{op}}(|\Delta^n|, T)$
- For $g \, : \, [n] \to [m]$ in Δ, $\mathrm{Sing}(g) \, : \, \mathrm{Sing}_m(T) \to \mathrm{Sing}_n(T)$ is given by precomposition with g_* as defined above.

We now want a way to go back. First, observe that we can have a mapping $[n] \mapsto |\Delta^n|$ by labelling the vertices of $|\Delta^n|$. To extend this, we recall one more fact about presheaf categories:

2.2.15 Theorem (Density Theorem). *Every presheaf is a colimit of representables. Concretely, given a small category $\mathcal{D}$ and a functor $F \, : \, \mathcal{D}^{op} \to \mathcal{S}\mathrm{et}$,*

$$F \cong \mathrm{colim}(\mathcal{D}_{/_F} \to \mathcal{D} \xrightarrow{y} \mathrm{Fun}(\mathcal{D}^{op}, \mathcal{S}\mathrm{et}))$$

Applying the density theorem to *s*Set results in an isomorphism

$$X \cong \mathrm{colim}_{\Delta^n \to X \text{ in } \Delta_{/_X}} \Delta^n$$

for any simplicial set X.

2.2.16 Definition. Let X be a simplcial set. Its ***geometric realization*** is the topological space

$$|X| := \mathrm{colim}_{\Delta^n \to X \text{ in } \Delta_{/_X}} |\Delta^n|$$

In other words, we are extending the functor that sends Δ^n to $|\Delta^n|$ to colimits of this particular form.

2.2.17 Proposition. *The geometric realization functor is left adjoint to the singular simplicial set.*

Proof. We look at the induced maps on hom-sets:

$$\mathrm{Hom}_{\mathcal{T}\mathrm{op}}(|X|, Y) \cong \lim_{\Delta^n \to X \text{ in } \Delta_{/X}} \mathrm{Hom}_{\mathcal{T}\mathrm{op}}(|\Delta^n|, Y)$$

$$\mathrm{Hom}_{s\mathrm{Set}}(X, \mathrm{Sing}(Y)) \cong \lim_{\Delta^n \to X \text{ in } \Delta_{/X}} \mathrm{Hom}_s\mathrm{Set}(\Delta^n, \mathrm{Sing}(Y))$$

and the limits on the right are naturally isomorphic. □

2.2.18 Proposition. *Let X be a simplicial set. Then $|X|$ is a CW-complex.*

Proof sketch. One can define the simplicial sets $\mathrm{sk}_n(X)$ by iteratively gluing copies of Δ^n and prove that their geometric realization defines a CW-filtration (see [Lur18, Tag 0010] for a detailed discussion of the skeletal filtration). □

Now that we have this correspondence, we shift our focus to two more special simplicial sets.

2.2.19 Definition. Let $n \in \mathbb{N}$.

1. We define the ***boundary*** of Δ^n to be the simplicial set $\partial\Delta^n$ with its set of k-simplices being the set of non-surjective maps $[k] \to [n]$.
2. For $0 \leq i \leq n$, we define the i-th ***horn*** to be the simplicial set Λ_i^n with its set of k-simplices being the set of maps $[k] \to [n]$ for which there exists a $j \neq i$ that is not contained in their image.

We call the horns defined for $1 < i < n-1$ the ***inner horns***.

In other words, the boundary is the union of all faces and the i-th horn is the union of all faces except the i-th one.

Since the standard n-simplex contains all of the maps considered above, we have inclusions

$$\Lambda_i^n \hookrightarrow \partial\Delta^n \hookrightarrow \Delta^n$$

Finally, to motivate the definitions that we have been building up to for this section, let us study how these simplicial sets behave using some familiar constructions from before. In particular, we are interested in the following question: Knowing that a map from Δ^n to X is equivalently an element of X_n, what can the maps from Λ_i^n to X be said to represent?

2.2.20 Example. Consider the nerve $N\mathcal{C}$ of a small category $\mathcal{C}$. Unfolding the definition of the horn, a map from Λ_i^n is really a collection of $n-1$-simplices with compatibility conditions wherever they are connected. In the case of the nerve, an $n-1$-simplex is a chain of $n-1$ composable morphisms in $\mathcal{C}$. We split into two cases:

Assume that $1 \leq i \leq n-1$. The missing i-th face is now the composite of the morphisms in that place. Thus, we can always extend a map from Λ_i^n to $N\mathcal{C}$ to a full n-simplex of $N\mathcal{C}$!

Now assume that i is 0 or n. What happens now is that instead of trying to fill in the composite, we are given one morphism and the possible "composite" and we are trying to fill in the gap. There is one case where this is always possible: $\mathcal{C}$ being a groupoid.

In this example, note that the extension of an inner horn map to a simplex is unique: A category has unique compositions.

2.2.21 Definition. Let X be a simplicial set.

1. X is a ***Kan complex*** if all horn maps admit a filling, i.e., a map completing a commutative triangle
$$\begin{tikzcd}\Lambda^n_i \arrow[r] \arrow[d, hook] & X \\ \Delta^n \arrow[ur, dotted, "\exists"] & \end{tikzcd}$$
2. X is a ***quasicategory*** if the above property is satisfied for all inner horns.

The Homotopy Theory of Kan Complexes

The idea behind this section can be summarized like this: *Kan complexes model homotopy types.* The exposition for the first part is mainly based on [Lan21].

2.2.22 Proposition. *Let Y be a topological space. Then* $\operatorname{Sing}(Y)$ *is a Kan complex.*

Proof sketch. Using the adjunction from (2.2.17), solve the analogous lifting problem in $\mathcal{T}\mathrm{op}$. ☐

2.2.23 Definition. Let $f, g : K \to X$ be morphisms of simplicial sets. A ***simplicial homotopy*** from f to g is a map $H : K \times \Delta^1 \to X$ that restricts to f and g on the two endpoints of Δ^1, i.e. $H|_{\partial\Delta^1} = (f, g)$.

2.2.24 Proposition. *Let X be a Kan complex. Then simplicial homotopy defines an equivalence relation on* $\operatorname{Hom}_{s\mathrm{Set}}(K, X)$ *for any simplicial set K.*

Proof. See e.g. [GJ99, I.6, I.7]. ☐

2.2.25 Definition. Let (X, x_0) be a pointed Kan complex, meaning that $x_0 : \Delta^0 \to X$. For $n \geq 1$ we define its ***simplicial homotopy groups*** as sets of pointed homotopy classes

$$\pi_n(X, x_0) := [(\Delta^n, \partial\Delta^n), (X, x_0)]_*$$

2.2.26 Proposition. *Simplicial homotopy groups of Kan complexes are indeed groups for $n \geq 1$ and are abelian for $n \geq 2$.*

2.2.27 Remark. We can also define the set of path components of any simplicial set X as $\pi_0(X) = \operatorname{colim}\left\{ X_1 \overset{\partial^1_1}{\underset{\partial^1_0}{\rightrightarrows}} X_0 \right\}$ which fits into the same definition, only without a group structure.

Topologically, the definition of simplicial homotopy groups matches one of the equivalent definitions of the homotopy groups of a pointed space, seen as $[([0,1]^n, \partial[0,1]^n), (Y, y_0)]_*$. In fact:

2.2.28 Theorem. *The simplicial homotopy groups of a pointed Kan complex coincide with the homotopy groups of its geometric realization.*

We now describe how the objects we have constructed fit in two nice model structures, the latter of which will be very important in justifying how we treat both quasi-categories and complete Segal spaces as equivalent notions of "∞-categories".

2.2.29 Definition. A map of simplicial sets $f : X \to Y$ is a (inner) ***Kan fibration*** if any commutative square of the form

$$\begin{tikzcd}
\Lambda^n_i \arrow[r] \arrow[d, hook] & X \arrow[d, "f"] \\
\Delta^n \arrow[r] \arrow[ur, dotted, "\exists ?"] & Y
\end{tikzcd}$$

admits a lift for all (inner) horn inclusions.

2.2.30 Proposition (Classical model structure on simplcial sets, [Qui67]). *There is a model structure on* $s\mathcal{S}\mathrm{et}$ *such that*

- *The fibrations are the Kan fibrations.*
- *The cofibrations are the maps that are levelwise injective.*
- *The weak equivalences are the maps that induce weak equivalences after applying the geometric realization.*

We denote this model structure by $s\mathcal{S}\mathrm{et}_{Quillen}$*.*

2.2.31 Proposition. *A Kan fibration $f : X \to Y$ is trivial if and only if every fiber*

$$\begin{tikzcd}
f^{-1}(y) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\
\Delta^0 \arrow[r, "y"'] & Y
\end{tikzcd}$$

is contractible.

Proof. See e.g. [Joy, Proposition 8.23]. ☐

2.2.32 Remark (The ∞-category of spaces). So far, we have not defined any ∞-category that is not a nerve, so we have not made use of the homotopical structure. The most notable example of a homotopically interesting ∞-category is given by the ***homotopy coherent nerve***, which takes a *simplicially enriched* category to an ∞-category. Applying the homotopy coherent nerve to the simplicially enriched category Kan of Kan complexes, we get the

∞-***category of spaces (or anima!)*** $\mathcal{S}$ (or $\mathcal{A}n$). This category is the "higher replacement" of the category of Sets, which we use for representable functors.

This is the category where a lot of modern homotopy theory takes place. In particular, note that by the facts we have mentioned so far, $\mathcal{S}$ has its own Whitehead theorem, making it a *hypercomplete topos*. Detailed discussions of everything said above can be found in [Lur09] or [Lan21] (for the homotopy coherent nerve).

2.2.33 Remark. Note that the condition for a Kan fibration is extending the same diagram used for the definition of a Kan complex. Thus if we use the terminal object $*$ as the target, we see that the fibrant objects in $s\mathcal{S}\mathrm{et}_{\mathrm{Quillen}}$ are precisely the Kan complexes.

This model structure allows us to make the relationship between simplicial sets and topological spaces precise. Recall the standard model structure on $\mathcal{T}\mathrm{op}$ with the usual weak equivalences, Serre fibrations as fibrations and retracts of relative cell complexes as cofibrations. Then we have:

2.2.34 Theorem ([Qui67]). *The adjoint functors*

$$\begin{tikzcd}\mathcal{T}\mathrm{op} \arrow[r, shift left, "\mathrm{Sing}(-)"] & s\mathcal{S}\mathrm{et} \arrow[l, shift left, "|-|"]\end{tikzcd}$$

induce a Quillen equivalence between the classical model structures on $\mathcal{T}\mathrm{op}$ *and* $s\mathcal{S}\mathrm{et}$*.*

We described a model structure centered on Kan complexes, and now we turn our attention to quasicategories:

2.2.35 Proposition (Joyal model structure, [Joy]). *There is a model structure on* $s\mathcal{S}\mathrm{et}$ *such that*

- *The cofibrations are the monomorphisms.*
- *The fibrant objects are precisely the quasicategories.*

We denote this model structure by $s\mathcal{S}\mathrm{et}_{Joyal}$*.*

The Reedy model structure on simplicial spaces

We now define a model structure that guarantees "good" homotopical properties for simplicial spaces.

2.2.36 Proposition (Reedy model structure). *There is a model structure on* $s\mathcal{S}$ *such that*

- *The fibrations are the maps* $f : X \to Y$ *for which the morphism of simplicial sets*

  $$\mathrm{Map}_{s\mathcal{S}}(F(n), X) \to \mathrm{Map}_{s\mathcal{S}}(\partial F(n), X) \times_{\mathrm{Map}_{s\mathcal{S}}(\partial F(n), Y)} \mathrm{Map}_{s\mathcal{S}}(F(n), Y)$$

  is a Kan fibration for all $n \geq 0$*.*

- *The weak equivalences are the maps that are levelwise weak equivalences in* $s\mathcal{S}\mathrm{et}_{Quillen}$*.*

2.2.37 Remark. Although the model structure on simplicial space is all that we need for the rest of this thesis, this is just a special case of a very general construction, with Δ being an example of a *Reedy category*. Reedy categories and the Reedy model structure originally appear in [Ree]. Not everything we will mention for the Reedy model structure on $s\mathcal{S}$ holds for model structures on arbitrary Reedy categories, also see [Hir03, chapter 15] for the general theory.

The following proposition allows us to get a better grip on the definition:

2.2.38 Proposition ([BR12, 3.15+4.5]). *The Reedy and injective model structures on $s\mathcal{S}$ coincide. Equivalently, the cofibrations in $s\mathcal{S}_{Reedy}$ are precisely the monomorphisms.*

What we are really interested in are the fibrant objects in this model structure. Unfolding the definition, we get that a simplicial space X is *Reedy fibrant* if the map

$$\mathrm{Map}_{s\mathcal{S}}(F(n), X) \to \mathrm{Map}_{s\mathcal{S}}(\partial F(n), X)$$

is a Kan fibration for all $n \geq 0$.

We finish with some fundamental properties.

2.2.39 Lemma. *Let X be a Reedy fibrant simplicial space. Then:*

1. *The "source-target" map $X_1 \xrightarrow{(d_1,d_0)} X_0 \times X_0$ is a Kan fibration.*
2. *X_n is a Kan complex for all $n \in \mathbb{N}$.*

Proof sketch. (1.) Recall that $\partial\Delta^1$ is just two points, so this is precisely the resulting Kan fibration in the Reedy fibrancy condition.

(2.) Work inductively: For $n = 0$ we get that X_0 is Kan fibrant (the same thing as a Kan complex), and for the induction steps use that compositions and pullbacks of Kan fibrations are Kan fibrations.

☐

2.2.40 Remark. Note that the projection $X_0 \times X_0 \to X_0$ is also a Kan fibration, so we get that d_0 and d_1 are also Kan fibrations.

2.2.41 Proposition ([Rez01, Proposition 2.5]). *The Reedy model structure is cartesian closed: If $i : A \to B$ is a cofibration and $f : X \to Y$ is a fibration, then the induced map*

$$Y^B \to Y^A \times_{X^A} X^B$$

is a fibration, and additionally a weak equivalence if either i or f is.

In particular, X^B is Reedy fibrant for any Reedy fibrant X and any simplicial space B.

2.3. Segal Spaces

The Segal Maps

Let $W \in s\mathcal{S}$. We want to relate the higher levels of W to W_1. Fix an $n \in \mathbb{N}$. For $0 \leq i \leq n-1$ Consider the maps

$$\begin{aligned} \alpha^i \, : \, [1] &\to [n] \\ 0 &\mapsto i \\ 1 &\mapsto i+1 \end{aligned}$$

in Δ. Note that these fit into a commutative diagram

$$\begin{tikzcd}
 & & & {[n]} & & & \\
{[1]} \arrow[urrr, "\alpha^0"] & {[0]} \arrow[l, "1"'] \arrow[r, "0"'] & {[1]} \arrow[ur, "\alpha^1"] & \ldots \arrow[l, "1"'] \arrow[r, "0"'] & {[1]} \arrow[ul, "\alpha^{n-2}"'] & {[0]} \arrow[l, "1"'] \arrow[r, "0"'] & {[1]} \arrow[ulll, "\alpha^{n-1}"']
\end{tikzcd}$$

so the maps of $\alpha_i := (\alpha^i)^* : W_n \to W_1$ induce a morphism of simplicial sets

$$\varphi_n : W_n \to W_1 \times_{W_0} \cdots \times_{W_0} W_1$$

where $W_1 \times_{W_0} \cdots \times_{W_0} W_1$ is the limit of the diagram

$$W_1 \xrightarrow{d_1} W_0 \xleftarrow{d_0} W_1 \xrightarrow{d_1} \ldots \xleftarrow{d_0} W_1 \xrightarrow{d_1} W_0 \xleftarrow{d_0} W_1$$

induced from the horizontal level of the above diagram in Δ. The maps φ_n we just constructed are called the ***Segal maps***.

To understand what these maps try to capture, let us look at the case of the *discrete nerve* of a small category (note that this turns out to not satisfy all the homotopical conditions we will specify later, but it provides good intuition for the structure we are looking for).

2.3.1 Example. Let $\mathcal{C}$ be a small category and define its discrete nerve $N\mathcal{C}$ by

$$N\mathcal{C}_{kl} := \mathrm{Fun}([k], \mathcal{C})$$

i.e., Then $N\mathcal{C}_k$ contains commutative diagrams of k composable morphisms. We can then see that $N\mathcal{C}_1 \times_{N\mathcal{C}_0} \cdots \times_{N\mathcal{C}_0} N\mathcal{C}_1$ consists of exactly the same data and the Segal maps are bijections: d_0 and d_1 are the source and target maps, so this limit again contains composable morphisms.

We want homotopical data to be preserved in a similar way, and thus we reach the definition of a Segal space:

2.3.2 Definition. Let W be a Reedy fibrant simplicial space. W is called a ***Segal space*** if the Segal maps

$$W_n \xrightarrow{\varphi_n} W_1 \times_{W_0} \cdots \times_{W_0} W_1$$

are weak equivalences for all $n \geq 2$.

Why (not) homotopy pullbacks?

What we are trying to define is an object in which we can do both category theory and homotopy theory. Thus the fact that the definition of a Segal space contains a strict pullback diagram raises some concerns:

2.3.3 Example. For spaces, the notion of a limit is not homotopy-invariant. In the diagram

$$\begin{tikzcd}
& * \arrow[rr] \arrow[dd] & & * \arrow[dd] \\
\varnothing \arrow[ur, dotted] \arrow[rr] \arrow[dd] & & I \arrow[ur, "\sim"] \arrow[dd] & \\
& * \arrow[rr] & & * \\
\{2\} \arrow[ur, "\sim"] \arrow[rr] & & \mathbb{R} \arrow[ur, "\sim"] &
\end{tikzcd}$$

the front and the back squares are pullbacks, but $\varnothing \not\simeq *$.

In this situation, an adequate replacement for these strict fibers would be the homotopy fiber, defined for a map $f : X \to Y$ as the pullback

$$\begin{tikzcd}
\mathrm{hofib}_x(f) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & P(f) \arrow[d, "ev_0"] \\
* \arrow[r, "x"] & Y
\end{tikzcd}$$

where $P(f) := X^I \times_X Y$, this time along the ev_1 map.

We now want to do the same for limits in $s\mathcal{S}$. In general, we can define the concept of a ***homotopy (co)limit*** in a model category: A homotopy (co)limit of a diagram should be a version of a (co)limit that is invariant under natural transformations of diagrams that are objectwise equivalences.

2.3.4 Remark. The universal property should be similarly adjusted such that it is satisfied, up to homotopy, for diagrams that are compatible up to homotopy (see e.g. [Dug08, Section 9.5] for a discussion in simplicial topological spaces). For arbitrary model categories, we can define the homotopy limit functor to be the derived functor of the limit functor $\lim : \mathcal{C}^I \to \mathcal{C}$ (if it exists), as in [Rie14, Chapter 5]. Note: this means that the homotopy limit is not just a limit in the homotopy category.

The difficulty of working with structures like limits in a homotopy-coherent way is one of the reasons that working with ∞-categories, where everything works up to homotopy by definition, is so useful. In turn, we will see homotopy type theory replicates this behaviour using identity types, without needing the difficult constructions one encounters in the theory of quasicategories.

For our Segal space W, this is where Reedy fibrancy comes in. We have the following fact:

2.3.5 Proposition. *Let $A \to C \leftarrow B$ be a diagram in a model category $\mathcal{C}$ such that A, B, C are fibrant and one of the two morphisms is a fibration. Then the pullback $A \times_C B$ is also a homotopy pullback.*

Proof. See e.g. [Dug08, Chapter 14] for the dual proposition for homotopy colimits. □

Apply this to the Segal maps: If W is Reedy fibrant, then W_1 and W_0 are Kan complexes, i.e., Kan fibrant, and the maps $d_1, d_0 : W_1 \to W_0$ are fibrations (2.2.39), so the iterated pullback $W_1 \times_{W_0} \cdots \times_{W_0} W_1$ is actually a homotopy pullback!

2.3.6 Remark (The nerve theorem in *s*Set)**.** On simplicial sets, the Segal condition is not very interesting; it is an observation going back to Grothendieck that, because of the lack of higher structure on the levels themselves, a simplicial set satisfying the Segal condition is isomorphic to the nerve of a small category.

Morphisms and Homotopies

Let W be a Segal space. Since our aim is to provide a model for a higher category, we first need to define, inside W:

- A set of *objects* in W.
- For any two objects, a space of *morphisms* between them.
- For tuples of morphisms with compatible sources and targets, *compositions* (unique up to some notion of homotopy) of those morphisms.

Let us begin with objects and morphisms. In a simplicial set X, the set of objects would be X_0. Here, we apply this to the simplicial set at the 0-th level of W.

2.3.7 Definition. Define the set of ***objects*** in W to be the set $\mathrm{Obj}(W) \coloneqq W_{00}$.

For morphisms, note that an object, i.e., an element of W_{00}, is, equivalently, a map of simplicial sets $\Delta^0 \to W_0$. Now, let $x, y \in \mathrm{Obj}(W)$. We similarly represent (x, y) as a map $(x, y) : \Delta^0 \to W_0 \times W_0$. Thinking of W_1 as the space of all possible morphisms in W, we just want to pick out all the morphisms that have source x and target y.

2.3.8 Definition. We define the ***mapping space*** $\mathrm{Map}_W(x, y) \in s\mathrm{Set}$ to be the pullback

$$\begin{tikzcd}
\mathrm{Map}_W(x,y) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & W_1 \arrow[d, "{(d_1,d_0)}"] \\
\Delta^0 \arrow[r, "{(x,y)}"'] & W_0 \times W_0
\end{tikzcd}$$

2.3.9 Proposition. *For any Reedy fibrant simplicial space W and $x, y \in \mathrm{Obj}(W)$, the mapping space $\mathrm{Map}_W(x, y)$ is a Kan complex.*

Proof. As we saw in (2.2.39), Reedy fibrancy of W implies that $W_1 \to W_0 \times W_0$ is a Kan fibration. For any $0 \le i \le k$, consider the diagram

$$\begin{tikzcd}
\Lambda^k_i \arrow[r] \arrow[d] & \mathrm{Map}_W(x,y) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & W_1 \arrow[d, "{(d_1,d_0)}"] \\
\Delta^k \arrow[r] \arrow[ur, dotted, "\exists ?"'] & \Delta^0 \arrow[r, "{(x,y)}"'] & W_0 \times W_0
\end{tikzcd}$$

Since pullbacks preserve fibrations, there exists a lift $\Delta^k \to \mathrm{Map}_W(x,y)$. $\square$

Having a mapping space (as opposed to just a mapping set) provides us with a way to express (higher) homotopical relations in the following way:

We think of the set $\mathrm{Map}_W(x,y)_0$ as the set of *maps* between x and y. Then $\mathrm{Map}_W(x,y)_1$ corresponds to the set of *homotopies* between these maps, $\mathrm{Map}_W(x,y)_2$ to homotopies between homotopies, and so on for all higher levels. We now focus on the first two and formulate the notion of homotopy in a Segal space.

2.3.10 Definition. The ***homotopy category*** $\mathrm{Ho}(W)$ of a Segal space W is the category with

- $\mathrm{Obj}(\mathrm{Ho}(W)) = \mathrm{Obj}(W) = W_{00}$
- $\mathrm{Hom}_{\mathrm{Ho}(W)}(x,y) = \pi_0(\mathrm{Map}_W(x,y))$

and with composition defined by the Segal map $W_2 \xrightarrow{\sim} W_1 \times_{W_0} W_1$. For $x \in \mathrm{Obj}(W)$, we define the ***identity*** $id_x := [s_0(x)]$, where $s_0 : W_{00} \to W_{10}$ is the degeneracy map.

We say that two maps $f, g \in \mathrm{Map}_W(x,y)_0$ are ***homotopic*** ($f \sim g$) if they lie in the same path component.

To understand this (and verify that $Ho(W)$ is actually a category), let us take a more detailed look at homotopies and compositions:

Recall that for a simplicial set X, we have

$$\pi_0(X) = \mathrm{colim}\left\{ \begin{tikzcd} X_1 \arrow[r, shift left, "d_1"] \arrow[r, shift right, "d_0"'] & X_0 \end{tikzcd} \right\}$$

which, applied to our mapping space, means that two maps f, g are homotopic if and only if there exists an $H \in \mathrm{Map}_W(x,y)_1$ with $d_1(H) = f,\ d_0(H) = g$.

We now show how to use the Segal condition to find compositions of tuples of maps. Using the Segal maps and the fact that we are pulling back along fibrations, we can define generalized mapping spaces and relate them to spaces of compatible maps as sketched in the diagram

$$\begin{tikzcd}
& \mathit{Map}_W(x_0,x_1) \times \cdots \times \mathit{Map}_W(x_{n-1},x_n) \arrow[rr] \arrow[ddl] \arrow[dr, phantom, "\lrcorner", very near start] & & W_1 \times_{W_0} \cdots \times_{W_0} W_1 \arrow[ddl] \\
\mathit{Map}_W(x_0,\dots,x_n) \arrow[ur, "\sim"] \arrow[rr] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & & W_n \arrow[ur, "\sim"] \arrow[d] & \\
\Delta^0 \arrow[rr] & & W_0^{n+1} &
\end{tikzcd}$$

Specialize this to $n = 2$: For any $x, y, z \in W_{00}$ and two maps $f : x \to y,\ g : y \to z$, there is a map from the ***space of compositions***

$$\mathrm{Comp}_W(f,g) := \mathrm{Map}_W(x,y,z) \times_{\mathrm{Map}_W(x,y) \times \mathrm{Map}_W(y,z)} \Delta^0 \to \mathrm{Map}_W(x,z)$$

induced by $d_1 : W_2 \to W_1$. This picks a composition of f and g. Is this choice arbitrary?

2.3.11 Proposition. *Composition is unique up to homotopy (i.e., $\mathrm{Comp}_W(f,g)$ is contractible) and* $\mathrm{Ho}(W)$ *is a category.*

2.3.12 Remark. Segal objects can be more generally defined as simplicial objects in arbitrary combintatorial model categories, using the same Segal condition as the starting point (see e.g. [DK19, Chapter 5]).

We call $f \in \mathrm{Map}_W(x,y)_0$ a ***homotopy equivalence*** if it is an isomorphism in $\mathrm{Ho}(W)$, i.e., if it is *invertible up to homotopy*. Already from ordinary category theory, we know that f has a homotopy inverse if and only if it has a left and a right homotopy inverse.

2.3.13 Lemma ([Rez01, Lemma 5.8])**.** *Any vertex that is connected to a homotopy equivalence is a homotopy equivalence.*

Taking advantage of this homotopical structure, we define, analogously to the case of simplicially enriched categories:

2.3.14 Definition ([Rez01, Definition 7.4])**.** A map $f : V \to W$ of Segal spaces is called a ***Dwyer-Kan equivalence*** if:

1. The induced map $\mathrm{Ho}(f) : \mathrm{Ho}(V) \to \mathrm{Ho}(W)$ is an equivalence of categories.
2. for any pair of objects x, y in V, the induced map $\mathrm{Map}_V(x,y) \to \mathrm{Map}_W(fx, fy)$ is a weak equivalence of simplicial sets.

2.4. Complete Segal Spaces

Let W be a Segal space and $x \in \mathrm{Obj}(W)$. Observe that by mapping x to $id_x \in \mathrm{Map}_W(x,x)_0$, we always get a homotopy equivalence. The motivating question is the following: Since we are doing mathematics up to homotopy, how do we make sure that this categorical construction treats, in some way, equivalence-as-equality?

In particular, we want a condition that translates the relationship of objects being connected by paths to them being equivalent.

Define the ***space of homotopy equivalences*** W_{hoequiv} to be the subspace of W_1 consisting of all components whose points (i.e., 0-simplices) are homotopy equivalences.

The degeneracy s_0 induces a map $s_0 : W_0 \to W_{\mathrm{hoequiv}}$. The question for before can now be reformulated in the following definition:

2.4.1 Definition. A Segal space W is ***complete*** if $s_0 : W_0 \to W_{\mathrm{hoequiv}}$ is a weak equivalence of simplicial sets.

Complete Segal spaces (finally!) exhibit all the desirable homotopical and categorical properties we need. In fact, the homotopical structure defined in the previous section now determines the homotopy theory of the space, as is made precise by the following results.

2.4.2 Theorem. *Let W be a complete Segal space and let* $^{\mathrm{Obj}(W)}\!/\!_{\sim}$ *denote the set of homotopy equivalence classes of objects in* $\mathrm{Ho}(W)$*. Then* $\pi_0(W_0) \simeq {}^{\mathrm{Obj}(W)}\!/\!_{\sim}$.

2.4.3 Theorem. *Let $f : V \to W$ be a map of complete Segal spaces. Then f is a Reedy equivalence if and only if it is a Dwyer-Kan equivalence.*

Recall that this is equivalent to f being a levelwise weak equivalence of simplicial sets. Moreover, there is a model structure on $s\mathcal{S}$ which is obtained as a particular localization of the Reedy model structure, with the following properties:

2.4.4 Theorem (Model Structure for Complete Segal Spaces [Rez01, Proposition 7.2])**.** *There is a simplicial model structure on $s\mathcal{S}$ such that*

- *The cofibrations are the monomorphisms.*
- *The fibrant objects are the Complete Segal Spaces.*
- *The weak equivalences are the maps f such that $\mathrm{Map}_{s\mathcal{S}}(f, W)$ is a weak equivalence of simplicial sets for any complete Segal space W.*

A Reedy weak equivalence is always an equivalence in this model structure. For a map of complete Segal spaces, the converse also holds.

Examples

2.4.5 Example. Let X be a Kan complex. As this is our notion of ”space”, there should be a sensible way of turning this into a complete Segal space. Note that just turning X into a constant simplicial space does not work: Reedy fibrancy would require $X \to X \times X$ to be a Kan fibration, which is not true.

We can, however, turn to a familiar topological construction: Recall that maps admit a path space factorization into a weak equivalence followed by a fibration. We thus define the n-th level of our simplicial space to be X^{Δ^n}. Since Δ^n is always contractible, X^{Δ^n} is always equivalent to X and additionally, $\{X^{\Delta^n}\}_n$ is now a (Reedy fibrant) complete Segal space.

Note that since X is a Kan complex, every map in this Segal space is a homotopy equivalence, making our space a ***Segal groupoid***. In fact:

2.4.6 Proposition ([Rez01, Proposition 6.6])**.** *Let W be a complete Segal space. Then* $\mathrm{Ho}(W)$ *is a groupoid if and only if W is Reedy weakly equivalent to a constant simplicial space.*

2.4.7 Example (CSS out of a Category)**.** Let $\mathcal{C}$ be a (small) category. The discrete nerve $N\mathcal{C}$ of $\mathcal{C}$, taken to be the nerve simplicial set on every level, does not satisfy the completeness condition, as the categorical equivalence of two objects inside $\mathcal{C}$ does not always translate to a homotopical equivalence (or example, consider the nerve of the category with two objects and isomorphisms between them).

To get around this problem, we have to take equivalences into account at the vertical level as well. With this intuition in mind, we outline nerve construction, the ***classifying diagram*** (or Rezk nerve), denoted $\mathcal{N}\mathcal{C}$.

The classifying diagram is defined as a simplicial space by

$$\mathcal{N}\mathcal{C}_m := N(\mathrm{Fun}([m], \mathcal{C})^{\simeq})$$

For a more explicit description (now as a bisimplicial set), let $I[n]$ be the category with $n+1$ objects and equivalences between them. Then

$$(\mathcal{N}\mathcal{C})_{mn} = Fun([m] \times I[n], \mathcal{C})$$

Using this, we sketch the proof that $\mathcal{N}\mathcal{C}$ is a complete Segal space:

We have $\mathcal{N}\mathcal{C}_0 = N((\mathcal{C}^{[0]})^{\simeq}) \cong N(\mathcal{C}^{\simeq})$, an equivalence of categories $\mathcal{C}^{\simeq} \xrightarrow{\sim} Fun(I[1], \mathcal{C})^{\simeq}$ and $\mathcal{N}\mathcal{C}_{hoequiv} \cong N(Fun(I[1], \mathcal{C})^{\simeq})$. Combining these, $\mathcal{N}\mathcal{C}_0 \xrightarrow{\sim} \mathcal{N}\mathcal{C}_{hoequiv}$.

2.4.8 Remark. We can repeat this construction in a more general setting, where we replace the groupoid core by an appropriate class of "weak equivalences" (for example, weak equivalences in a model category). The resulting nerve will not be a complete Segal space in general, but its Reedy fibrant replacement is! [Rez01, 8.3]

2.4.9 Proposition. *Let $F : \mathcal{C} \to \mathcal{D}$ be a functor. The following are equivalent:*

1. *F is an equivalence of categories.*
2. *$\mathcal{N}F : \mathcal{N}\mathcal{C} \to \mathcal{N}\mathcal{D}$ is a Reedy equivalence.*

2.4.10 Example (CSS out of a quasicategory). Finally, we can present the comparison between quasicategories and complete Segal spaces, originally due to Joyal and Tierney [JT07]. As we mentioned in the first section, comparing these models means comparing the appropriate model structures, in our case the ones defined in (2.2.35) (for quasicategories) and in (2.4.4) (for complete Segal spaces). We will define two adjoint pairs:

Consider the functor $i_1 : \Delta \to \Delta \times \Delta$ given by $[n] \mapsto ([n], [0])$. Dualizing, this turns into a functor

$$i_1^* : s\mathcal{S} \to s\mathrm{Set}\,, \quad (i_1^*(W))_n := W_{n0}$$

Now do the same for the right adjoint of i_1, the first projection functor $p_1 : \Delta \times \Delta \to \Delta$ given by $([m], [n]) \mapsto [m]$. Passing to presheaves again, we now get a left adjoint to i_1^*, the functor

$$p_1^* : s\mathrm{Set} \to s\mathcal{S}\,, \quad (p_1^*(X))_{mn} := X_m$$

with the adjunction unit being the identity functor on $s\mathrm{Set}$ and the counit being the inclusion (via degeneracies) $W_{m0} \hookrightarrow W_{mn}$.

2.4.11 Theorem ([JT07]). *p_1^* maps quasicategories to complete Segal spaces, i_1^* maps complete Segal spaces to quasicategories, and the adjoint pair*

$$\begin{tikzcd} s\mathrm{Set} \arrow[r, shift left, "p_1^*"] \arrow[r, phantom, "\perp"] & s\mathcal{S} \arrow[l, shift left, "i_1^*"] \end{tikzcd}$$

is a Quillen equivalence between the Joyal model structure for quasicategories and the model structure for complete Segal spaces.

Moreover, there is another Quillen equivalence

$$\begin{tikzcd} s\mathcal{S} \arrow[r, shift left, "t_!"] \arrow[r, phantom, "\perp"] & s\mathrm{Set} \arrow[l, shift left, "t^!"] \end{tikzcd}$$

in the opposite direction.

3. Homotopy Type Theory

This chapter is intended to be an expository introduction to the basic mechanics of homotopy type theory, starting from scratch. We first introduce dependent type theory and focus on the path space interpretation of intensional identity types, drawing analogies with topology and the higher structure of Kan complexes as defined in the previous chapter.

As we will be working with formal mathematics, we start with an axiomatic approach, and also devote some sections to the logical aspects of homotopy type theory: In its full version with univalence and higher inductive types, it is a fundamentally constructive theory, allowing us to do classical logic after truncating types to "forget the higher structure".

3.1. Syntax and Axioms of Martin-Löf Type Theory

In this section, we present an overview of the basic Syntax of a specific type theory. To specify a system of type theory, we need a collection of ***structural rules*** and ***type-forming rules***. The type theory that is of interest to us is ***Martin-Löf dependent type theory*** (MLTT). For a complete formal presentation of MLTT (and HoTT), we point to [Uni13, Appendix A].

3.1.1 Definition.

- A ***context*** is a finite list of *declarations* of the form $x_i : A_i$. We add an empty context () with no declarations.
- A ***judgement*** is of the form $\Gamma \vdash \mathcal{J}$, where Γ is a context. In MLTT, the possible forms of judgements are

$$\Gamma \ \mathsf{ctx} \qquad\qquad \Gamma \vdash a : A \qquad\qquad \Gamma \vdash a \equiv a' : A$$

In a context, the x_i are *variables*, but in the judgement $\Gamma \vdash a : A$, a is a specific term of type A that can depend on the declarations in Γ.

Note the symbol "$\equiv$" : This expresses the notion of ***judgemental equality***, which will be the formal specification of *equality by definition*. We will revisit this later, as we will also present a different notion of equality, having inhabited identity types. Whether or not these coincide is the difference between *extensional* and *intensional* type theories.

3.1.2 Notation. To specify the allowed steps in type-theoretic constructions and proofs, we express our ***inference rules*** in the form

$$\frac{\mathcal{J}_1 \ \dots \ \mathcal{J}_n}{\mathcal{C}}$$

where all the $\mathcal{J}_i$ and $\mathcal{C}$ are judgements.

Structural Rules

First of all, consider the context $a : A$. We want A itself to have a type. We thus assume a hierarchy of ***universes*** $\mathcal{U}_i$, $i \in \mathbb{N}$, with the extra assumption that the universes are *cumulative*, i.e., a type in a universe is also in the next one. In formal terms, we have two rules:

$$\frac{\Gamma \text{ ctx}}{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}} \qquad\qquad \frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash A : \mathcal{U}_{i+1}}$$

3.1.3 Notation. Usually, We will not need the entire hierarchy to do our type constructions. When all of our types are in the same universe, we just use some fixed universe denoted by $\mathcal{U}$.

3.1.4 Remark. In other formulations of Type Theory, we might see the additional judgements $\Gamma \vdash A$ type and $\Gamma \vdash A \equiv B$ type. This is rendered obsolete if we work with universes from the start, as they can replaced by $\Gamma \vdash A : \mathcal{U}$ and $\Gamma \vdash A \equiv B : \mathcal{U}$ respectively.

0. **Rules for contexts**. We already gave a definition of a context, and now we can make it precise with three inference rules. For the first part, we need to say that we can introduce both the empty context and the extension of a finite number of declarations with one more variable:

$$\frac{}{() \text{ ctx}} \qquad\qquad \frac{x_1 : A_1, \dots, x_n : A_n \vdash A_{n+1} : \mathcal{U}}{(x_1 : A_1, \dots, x_{n+1} : A_{n+1}) \text{ ctx}}\text{ctx} - \text{EXT}$$

We then state that we can actually use the declarations in the context, or, in other words, that they genuinely capture variables:

$$\frac{(x_1 : A_1, \dots, x_n : A_n) \text{ ctx}}{x_1 : A_1, \dots, x_n : A_n \vdash x_i : A_i}\text{Vble}$$

Before we deal with any actual type constructions, we have to provide rules outlining the basic structure of our system. It is important to note that, unlike classical logic, the deductive system of type theory is on the same level as the specified type theory itself, so we now have to ensure that everything works in a "sensible" way. We split these rules in the following categories:

1. **Rules for judgemental equality**. We start in a similar way to any axiomatization of classical logic and set theory: Equality is an equivalence relation. We can express reflexivity, symmetry and transitivity as we usually would:

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv a : A} \qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b \equiv a : A} \qquad \frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash b \equiv c : A}{\Gamma \vdash a \equiv c : A}$$

but there is more that we have to say: Since everything is now done inside some type, we also have to state that judgemental equality between terms is preserved by judgemental equality between types:

$$\frac{\Gamma \vdash A \equiv B : \mathcal{U} \quad \Gamma \vdash a : A}{\Gamma \vdash a : B} \qquad \frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash A \equiv B : \mathcal{U}}{\Gamma \vdash a \equiv b : B}$$

2. **Variable conversion**. This describes the interplay between judgemental equality and variables. Namely, we can freely exchange between types that are judgementally equal:

$$\frac{\Gamma \vdash A \equiv A' : \mathcal{U} \quad \Gamma, x : A, \Delta \vdash \mathcal{J}}{\Gamma, x : A', \Delta \vdash \mathcal{J}}$$

3. **Substitution**. Here, we have rules for replacing a variable with a specific term, as we would in classical mathematics. We first require that that such substitutions are possible

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, \Delta \vdash b : B}{\Gamma, \Delta[a/x] \vdash b[a/x] : B[a/x]}$$

   and then that they respect judgemental equalities

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, \Delta \vdash b \equiv c : B}{\Gamma, \Delta[a/x] \vdash b[a/x] \equiv c[a/x] : B[a/x]} \qquad \frac{\Gamma \vdash a \equiv a' : A \quad \Gamma, x : A, \Delta \vdash b : B}{\Gamma, \Delta[a/x] \vdash b[a/x] \equiv b[a'/x] : B[a/x]}$$

4. **Weakening**. Finally, we say that we can add more declarations to a context, without affecting the judgement in any other way:

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma, \Delta \vdash b : B}{\Gamma, x : A, \Delta \vdash b : B} \qquad \frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma, \Delta \vdash b \equiv c : B}{\Gamma, x : A, \Delta \vdash b \equiv c : B}$$

Type Forming Rules

One can think of inference rules as in classical logic, but they are more general than that: They define all the possible ways we can produce elements in any given type that our axioms allow.

To provide the axioms for a specific type, we put our deductive rules into the following categories, each one specifying a different aspect of the behavior of our desired type. We have:

1. **Formation rules**, which specify the conditions under which our type exists.

2. **Introduction rules**, i.e., ways to construct terms in the type.

3. **Elimination rules**, describing how to use a term of the type.

4. **Computation rules**, describing how elimination rules act when applied to introduction rules.

5. **Uniqueness principles** (optionally), imposing uniqueness conditions on mapping in or out of the type. These usually involve judgmental equalities, but some types have provable uniqueness principles involving intensional identity types.

3.1.5 Remark. There is one set of axioms that we assume but do not explicitly state: With every intro rule comes a rule saying that it respects $\equiv$.

Going another way, we will see examples of how these rules can have direct analogues to universal properties in category theory.

In this section, we fix a universe $\mathcal{U}$. We begin with a fundamental definition which will add the ”dependent” in dependent type theory.

3.1.6 Definition. A ***type family*** over $A : \mathcal{U}$ in context Γ is of the form

$$\Gamma, x : A \vdash B(x) : \mathcal{U}$$

In plain language, this means that a type family assigns a type $C(x) : \mathcal{U}$ to every term of A.

3.1.7 Remark. If one has a working definition of a function type, then a type family is just a function $C : A \to \mathcal{U}$

We can now begin to axiomatically describe Martin-Löf Type Theory. Let us start with two very simple fundamental cases:

3.1.8 Example (Empty and unit type)**.** As the names suggest, we want to provide rules for a type analogous to the empty and one-point sets, respectively. To begin with, there is no reason for the existence of these types to depend on anything else, so the formation rules are simple, allowing any context:

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbf{0} : \mathcal{U}} \qquad\qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbf{1} : \mathcal{U}}$$

On the category-theoretic end, the appropriate generalization is the initial and terminal objects, characterized by mapping properties. Looking at our type-forming rules, this is where the elimination rules come in. Note that contexts play a role here: In categorical semantics of type theory, contexts themselves are interpreted as objects.

We first consider the empty type: We have an elimination rule

$$\frac{\Gamma, x : \mathbf{0} \vdash C : \mathcal{U} \quad \Gamma \vdash a : \mathbf{0}}{\Gamma \vdash \mathsf{ind}_{\mathbf{0}}(x.C, \alpha) : C[a/x]}$$

and no introduction or computation rules. This makes sense: There should be no way to construct a term in the empty type.

Let us now try to interpret this elimination rule: Assume we have a type A and a context Γ (i.e., the type family $x : \mathbf{0} \vdash C$ is the constant family A). Applying weakening, we have a well-formed judgement

$$\Gamma, x : \mathbf{0} \vdash A : \mathcal{U}$$

where A is a constant type family over $\mathbf{0}$. Then, applying the elimination rule we get a term in A for every term in $\mathbf{0}$, i.e., a function

$$\mathbf{0} \to A$$

Switching to a categorical perspective, this is part of the condition for $\mathbf{0}$ to be initial: there exist a map to any other object. Moreover, if we think of the empty type as representing the logical statement false, then we have the *ex falso* rule: Anything can be derived from false.

We now move to the unit type. For the introduction rule, the unit type comes with one term:

$$\frac{\Gamma \ \mathsf{ctx}}{\Gamma \vdash \star : \mathbf{1}}$$

For the elimination and computation rules, we want a description of how to map out of the type. Getting some intuition from set theory, we expect that the only thing we need to do is specify the term where $\star$ gets sent to. Indeed, we have an elimination rule

$$\frac{\Gamma, x : \mathbf{1} \vdash C : \mathcal{U} \quad \Gamma \vdash c_\star : C[\star/x] \quad \Gamma \vdash a : \mathbf{1}}{\Gamma \vdash \mathsf{ind}_\mathbf{1}(x.C, \alpha, c_\star) : C[a/x]}$$

saying that if we have a family of types parameterized by terms of $\mathbf{1}$, a term $c_\star$ corresponding to $\star$ and a term in $\mathbf{1}$, we can have an element $\mathsf{ind}_\mathbf{1}(x.C, \alpha, c_\star)$ in the appropriate type. What we have not yet said is that $\mathsf{ind}_\mathbf{1}(x.C, -, c_\star)$ is actually $c_\star$ when evaluated at $\star$. To do that, we have our first example of a computation rule

$$\frac{\Gamma, x : \mathbf{1} \vdash C : \mathcal{U} \quad \Gamma \vdash c_\star : C[\star/x]}{\Gamma \vdash \mathsf{ind}_\mathbf{1}(x.C, \star, c_\star) \equiv c_\star}$$

Similarly to $\mathbf{0}$ being our form of initial object, we would like $\mathbf{1}$ to somehow be terminal, or, in terms of logic, the statement true. That requires all types to map into the unit type, which we can do by taking everything to $\star$, but also requires some uniqueness condition. Instead of assuming that every term of $\alpha \equiv \star$ for every $\alpha : \mathbf{1}$ judgmentally as an axiom (a uniqueness rule for $\mathbf{1}$), we will later show that another form of such a statement is provable using intensional identity types. We will also see that this is reasonable under our homotopy interpretation, as uniqueness conditions in higher category theory are specified up to homotopy.

Dependent function types

Given a type family B over A, a term of the type $\prod_{x:A} B(x)$ assigns to every term $x : A$ a term of $B(x)$. We have a formation rule

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma, x : A \vdash B(x) : \mathcal{U}}{\Gamma \vdash \prod_{x:A} B(x) : \mathcal{U}}$$

and an introduction rule

$$\frac{\Gamma, x : A \vdash b : B(x)}{\Gamma \vdash \lambda(x : A).b : \prod_{x:A} B(x)}$$

stating formally what we just described. In particular, the rule $\lambda(x : A).b$ is our equivalent of $f(x) :\equiv b$, and we similarly want x to be bound in b.

The elimination rule is there to make sure that we always end up in the right type:

$$\frac{\Gamma \vdash f : \prod_{x:A} B(x) \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B[a/x]}$$

and the computation and uniqueness rules make precise what we said above: Assigning $b(x)$ to every x results in the function with these exact values.

$$\frac{\Gamma x : A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda(x : A).b)(a) \equiv b[a/x] : B[a/x]} \qquad \frac{\Gamma \vdash f : \prod_{x:A} B(x)}{\Gamma \vdash f \equiv \lambda x.f(x) : \prod_{x:A} B(x)}$$

3.1.9 Remark. Consider the case where the family is constant, i.e., a type B. Then, we obtain the ordinary *function type*

$$A \to B :\equiv \prod_{x:A} B$$

In our formal system, this will be the definition of the function type. In other presentations we can also define this function type without using families, so we can instead take the definition of a type family to be a term of type $A \to \mathcal{U}$.

3.1.10 Remark. Using the (ordinary) function type, we can express functions with multiple parameters: A function $f : A \times B \to C$ corresponds precisely to a function $f : A \to (B \to C)$. This operation is called ***currying*** and is the preferred way of denoting such types.

3.1.11 Notation. We usually skip the parenthesis on the right when there is no ambiguity, writing $A \to B \to C$ for $A \to (B \to C)$.

Dependent sum types

The “classical equivalent” of these would be an indexed disjoint union. Let’s take a look at the details: The formation rule works as with the dependent function type. Again, we highlight the introduction rule

$$\frac{\Gamma, x : A \vdash B(x) : \mathcal{U} \quad \Gamma \vdash a : A \quad \Gamma \vdash b : B[a/x]}{\Gamma \vdash (a, b) : \sum_{x:A} B(x)}$$

Note the difference from the rule of the $\prod$-type: Here, we only require some term $a : A$ to “produce” a term of B, and, if this is the case, we get a pair in the $\sum$-type.

The elimination rule

$$\frac{\Gamma, z : \sum_{x:A} B(x) \vdash C : \mathcal{U} \quad \Gamma, x : A, y : B(x) \vdash g : C[(x, y)/z] \quad \Gamma \vdash p : \sum_{x:A} B(x)}{\Gamma \vdash \mathsf{ind}_{\sum_{x:A} B(x)}(z.C, x.y.g, p) : C[p/z]}$$

and computation rule

$$\frac{\Gamma, z : \sum_{x:A} B(x) \vdash C : \mathcal{U} \quad \Gamma, x : A, y : B(x) \vdash g : C[(x, y)/z] \quad \Gamma \vdash a : A \quad \Gamma \vdash b : B[a/x]}{\Gamma \vdash \mathsf{ind}_{\sum_{x:A} B(x)}(z.C, x.y.g, (a, b)) \equiv g[(a, b)/(x, y)] : C[(a, b)/z]}$$

provide an inductive mapping property for $\sum$-types: if we have a family C over the $\sum$-type and ways to map into C from A and B, then this extends to a dependent function of type

$$\prod_{p:\sum_{x:A} B(x)} C(p)$$

We will see more such cases of inductive definitions interpreted in natural language, and will use them in this form to provide readable proofs. For example:

3.1.12 Construction (Projections). We use induction to define maps that single out the first and second coordinates of pairs respectively.

- The first projection $\mathsf{pr}_1 : (\sum_{x:A} B(x)) \to A$ is defined inductively by setting $\mathsf{pr}_1(a, b) :\equiv a$ on pairs.
- As B can be any family over A, the second projection has to be a dependent function, since the type we will end up in depends on the first coordinate. We define

  $$\mathsf{pr}_2 : \prod_{p:\sum_{x:A} B(x)} B(\mathsf{pr}_1(p))$$

  once again using induction and setting $\mathsf{pr}_2(a, b) :\equiv b$ on pairs.

3.1.13 Remark. As with the $\prod$-type and the ordinary function type, we have a case that falls out of the definition of $\sum$-types: If the family B is constant, then we define the ***product***

$$A \times B :\equiv \sum_{x:A} B$$

and, applying the previous construction to the constant case, we have the usual projection maps $\mathsf{pr}_1 : A \times B \to A$, $\mathsf{pr}_1(a, b) :\equiv a$ and $\mathsf{pr}_2 : A \times B \to B$, $\mathsf{pr}_2(a, b) :\equiv b$.

3.1.14 Remark. As the case of the product and the interpretation of the elimination and computation rules might hint at, we can define $\sum$-types as *inductive types* satisfying a form of the universal property of the product for dependent functions. This type of definition is what we use in practice, and it will become especially important in the case of identity types. This is also how we will define coproducts:

Coproducts

We saw that products exist in type theory as a special case of dependent sum types. Here, we present the axioms for the existence of coproduct types as well. As usual, we have a formation rule

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathcal{U}}{\Gamma \vdash A + B : \mathcal{U}}$$

Analogously to the coproduct in category theory, there should be inclusions $A \xrightarrow{\mathsf{inl}} A + B$ and $B \xrightarrow{\mathsf{inr}} A + B$, which are presented by the introduction rules

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathcal{U} \quad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{inl}(a) : A + B} \qquad \frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathcal{U} \quad \Gamma \vdash b : B}{\Gamma \vdash \mathsf{inr}(b) : A + B}$$

meaning that we can construct terms in $A + B$ in two ways, using terms in A or B.

The rest of the axioms for coproducts can be packaged into an inductive definition outlining the universal property of the coproduct:

Given a type family

$$\Gamma, x : A + B \vdash P(x) : \mathcal{U}$$

there is a function

$$\mathsf{ind}_+ : \Big(\prod_{a:A} P(\mathsf{inl}(a))\Big) \to \Big(\big(\prod_{b:B} P(\mathsf{inr}(b))\big) \to \prod_{x:A+B} P(x)\Big)$$

satisfying the computation rules

$$\mathsf{ind}_+(f, g, \mathsf{inl}(a)) \equiv f(a)$$
$$\mathsf{ind}_+(f, g, \mathsf{inr}(b)) \equiv g(b)$$

In natural language, the coproduct $A + B$ comes with a function that sends pairs of (dependent) functions out of A and B that are compatible with the respective inclusion to (dependent) functions out of $A + B$.

Recall that, in category theory the universal property of the coproduct requires that maps out of $A + B$ are uniquely determined by maps out of A and B that commute with the inclusions. The uniqueness part of the definition will again be propositional and not judgemental.

3.2. Logical aspects, part I: Curry-Howard

Since we are developing type theory as its own deductive system, there should be a way of expressing type-theoretic analogues to statements written in the language of propositional logic. In this section we describe the intuition behind the propositions-as-types interpretation, and later we will make the different "levels" of logic in our types precise, using propositional truncations.

- We begin with our notion of a true statement: Giving a proof in type theory means constructing a term in a corresponding type. Thus we can say that the type corresponding to $\top$ is $\mathbf{1}$ and the type corresponding to $\bot$ is $\mathbf{0}$.

  A type expressing a true statement is thus an inhabited type. By the mapping property determined by the type-forming rules for $\mathbf{1}$, a type A is inhabited precisely when there exists a function $\mathbf{1} \to A$.

- Functions work like implications: Providing a function $A \to B$ means constructing a term in B for every term in A. Thus, if A is inhabited and there exists $f : A \to B$, then B is inhabited.

  Applying this to $\mathbf{1}$ and $\mathbf{0}$, we get the usual properties of $\top$ and $\bot$ respectively: We have $\mathbf{0} \to A$ and $A \to \mathbf{1}$ for any A, just as every statement is implied by $\bot$ implies $\top$.

- We now create our negations: For a type A, set

  $$\neg A :\equiv (A \to \mathbf{0})$$

  Since there are no introduction rules for $\mathbf{0}$, we should not be able to produce a term in $\mathbf{0}$ given a term in A.

Note that we can prove $A \to \neg(\neg A)$: Under this interpretation, given $a : A$ and $f : A \to \mathbf{0}$, applying f we can produce $f(a) : \mathbf{0}$. This is one side of the law of excluded middle, but there is no reason why the other should hold:

$\neg(\neg A) \to A$ is saying that there is some $M : (A \to \mathbf{0}) \to \mathbf{0} \to A$, i.e., that having a function $g : (A \to \mathbf{0}) \to \mathbf{0}$ should be enough to produce a term in A. Assuming the full axiomatic system of homotopy type theory, this cannot be true!

- Continuing our motivation from category-theoretic mapping properties, we express $\wedge$ as $\times$ and $\vee$ as the coproduct $+$. To check that the properties are satisfied, we can look at the maps

$$\begin{tikzcd}
& A \times B \arrow[dl, "\mathsf{pr}_1"'] \arrow[dr, "\mathsf{pr}_2"] & & & & A + B & \\
A & & B & & A \arrow[ur, "\mathsf{inl}"] & & B \arrow[ul, "\mathsf{inr}"']
\end{tikzcd}$$

On dependent types

Type families as predicates. We have already interpreted types are propositions and functions as implications. From this viewpoint, a type family $\Gamma, x : A \vdash C(x) : \mathcal{U}$ assigns, to every term $x : A$, the statement "$C(x)$ is inhabited". We can now interpret dependent function and pair types:

- We start with $\prod$-types. Since an inhabited function type can be seen as a logical implication, we can revisit our explanation of the introduction rule and interpret it as such: For every $x : A$, we have a witness of $B(x)$. We have thus found our analogue of the statement: *for all* $x : A, B(x)$ *holds*, or $\forall x B(x)$.
- Now do the same for dependent sum types: This time, constructing a term of type $\sum_{x:A} B(x)$ means that we have provided at least one term of A such that there is a term in $B(x)$. We have the analogue of: *there exists some* $x : A$ *for which* $B(x)$ *holds*, or $\exists x B(x)$.

  Note that the $\sum$-type is (at least) the collection of all such pairs of terms $x : A$ and proofs of $B(x)$. The analogy here is, once again, not classical: We will need propositional truncations to do classical logic.

The type of booleans and decidability

Using the existence of the coproduct and the unit type, we now construct a type of booleans or "truth values": We set

$$\mathsf{bool} :\equiv \mathbf{1} + \mathbf{1}$$

For the sake of clarity, denote the two specified points by $\mathsf{false} : \mathsf{bool}$ and $\mathsf{true} : \mathsf{bool}$. Applying the inductive definition of the coproduct for a family $P : \mathsf{bool} \to \mathcal{U}$ results in a recursor

$$\mathsf{rec}_{\mathsf{bool}} : P(\mathsf{false}) \to \big(P(\mathsf{true}) \to \prod_{x:\mathsf{bool}} P(x)\big)$$

or, assigning a term to $\mathsf{false} : \mathsf{bool}$ and a term to $\mathsf{true} : \mathsf{bool}$ is enough to map out of the whole type. In other words, the induction principle for bool is just *case analysis*.

However, this does *not* mean that we can do case analysis for any desired type-theoretic construction! Assume we want to construct a function in the type $A \to B$ using case analysis on a predicate $P : A \to \mathcal{U}$. The strategy would be to use $\mathsf{rec}_{\mathsf{bool}} : B(\mathsf{false}) \to \big(B(\mathsf{true}) \to (\mathsf{bool} \to B)\big)$ in the case of the constant family B. Then, we need to find a way to construct a map $A \to \mathsf{bool}$, deciding whether a term $a : A$ gets sent to true if $P(a)$ or false if $\neg P(a)$. This is where we get stuck: There is no axiom that says such a function should exist for any predicate.

3.2.1 Definition. 1. A type X is ***decidable*** if $X + \neg X$ is inhabited, i.e., the law of excluded middle holds for X.

2. A family $P : X \to \mathcal{U}$ is decidable if $\prod_{x:X}(P(x) + \neg P(x))$.

Assuming P is decidable, case analysis can be performed: Given a dependent function $c : \prod_{x:A}(P(x) + \neg P(x))$, define

$$\tilde{c} : A \to \mathsf{bool}\,, \quad \lambda x.t_{P,x}(c(x))$$

where $t_{P,x} : (P(x) + \neg P(x)) \to \mathsf{bool}$ is defined by sending $y : P(x)$ to true and $y : \neg P(x)$ to false.

3.3. Identity types

It is now time to look at identifications *internal* to our type theory. The key conceptual difference is that an identity type contains *all possible identifications* of two terms of a given type (there can be more than one!). For two terms $x, y : A$ the identity type $x =_A y$ is defined inductively: We have a formation rule

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash x : A \quad \Gamma \vdash y : A}{\Gamma \vdash x =_A y : \mathcal{U}}$$

For the introduction rule, there is only one term we can construct by default in the identity type: The term $\mathsf{refl}_x : x =_A x$ corresponding to a trivial identification of a term with itself. Formally, the rule is

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash x : A}{\Gamma \vdash \mathsf{refl}_x : x =_A x}$$

If one checks the rest of the formal presentation, the elimination and computation rules are quite complicated. However, they are outlining a powerful defining property, called ***path induction***.

The statement is as follows [Uni13, 1.12.1]:

Assume we have a family

$$\Phi : \prod_{x:A} \prod_{y:A} (x =_A y) \to \mathcal{U}$$

and a (dependent) function

$$\varphi : \prod_{x:A} \Phi(x, x, \mathsf{refl_x})$$

Then there is a function

$$f : \prod_{x:A} \prod_{y:A} \prod_{(p:x=_A y)} \Phi(x, y, p)$$

that "extends" φ:

$$f(x, x, \mathsf{refl}_x) = \varphi(x)$$

In terms of proof tactics, this is saying: If I want to prove something for all $x, y : A$ and all paths between them, it is enough to consider the case in which both elements are x and the path is refl_x! Or, from an informal categorical perspective, we have lifts

$$\begin{tikzcd}
A \arrow[r] \arrow[d, hook] & E \arrow[d, two heads] \\
PA \arrow[r] \arrow[ur, dotted, "\exists"] & B
\end{tikzcd}$$

where PA is the path space corresponding to the "total identity type" and the map on the right is a fibration (this will make more sense using our interpretations in the next sections).

Warning: This is *not* saying that these are the only terms in the identity type. In fact, identity types can be complicated enough to allow us to reason about homotopy theory in HoTT. Of course, it is not immediately clear why we would actually want these types to have that kind of structure. This is the first point where the classical and the homotopical perspective diverge.

3.4. Identity Types and Homotopy Theory

We now present a topological interpretation (and some facts) to justify the statement that intensional identity types provide types in Martin-Löf type theory with a higher homotopical structure. Consider a term $p : x =_A y$ in some identity type. This first proposition states that we can really think of such terms as *paths*, in the sense that they satisfy the same fundamental properties that paths do (*up to homotopy*) in topological spaces, or, in the model we are going for, Kan complexes.

3.4.1 Proposition. *Let A be a type. There is a concatenation operator of type*

$$\prod_{x,y,z:A} (x =_A y \to (y =_A z \to x =_A z))$$

and an inverse operator of type

$$\prod_{x,y:A} (x =_A y \to y =_A x)$$

satisfying the relevant associativity, unit and inverse laws, with the unit being refl_x.

Proof sketch. We prove this by path induction: Both types are inhabited when $x \equiv y$ and the paths are refl_x. ☐

Let us now look at the type

$$\sum_{x,y:A} (x =_A y)$$

This is analogous to the concept of the ***free path space***, i.e. the path space of a topological space not bound by any endpoints. In such a space, any path is homotopic to the constant path at one of its endpoints. The analogous principle of *based path induction* corresponds to the behavior of the *based path space.*

We now shift our focus to functions. The next proposition states that there is a functorial way to apply a function to a path.

3.4.2 Proposition. *Let $f : A \to B$, $g : B \to C$ be functions. there is an operation (**action on paths**)*

$$\mathsf{ap}_f : \prod_{x,y:A} (x =_A y \to f(x) =_B f(y))$$

together with operations

$$\mathsf{ap\text{-}id}_A : \prod_{x,y:A} \prod_{(p:x=_A y)} p =_{(x=_A y)} \mathsf{ap}_{\mathsf{id}_A}(p)$$

$$\mathsf{ap\text{-}comp}(f,g) : \prod_{x,y:A} \prod_{(p:x=_A y)} \mathsf{ap}_g(\mathsf{ap}_f(p)) =_{(g\circ f(x) =_C g\circ f(y))} \mathsf{ap}_{g\circ f}(p))$$

Moreover, we have a dependent version: If $f : \prod_{x:A} C(x)$, then there is an operation

$$\mathsf{apd}_f : \prod_{(p:x=_A y)} (f(x) =_{C(x)} f(y))$$

Once again, the key here is to iteratively define the operators, each time in the case where the relevant path is refl, and use path induction.

We can have an analogous operator in the dependent function type, for which we need:

3.4.3 Proposition (Transport). *Let A be a type and C a type family over A. There is an operation*

$$\mathsf{tr}_C : \prod_{x,y:A} (x =_A y \to (C(x) \to C(y))$$

3.4.4 Notation. Let $p : x =_A y$. We denote by $p_* : (C(x) \to C(y))$ the function $\mathsf{tr}_C(x, y, p)$.

3.4.5 Definition. A type A is called ***contractible*** if the type

$$\mathsf{isContr}(A) :\equiv \sum_{x:A} \prod_{y:A} (y =_A x)$$

is inhabited.

Unfolding this definition, we can see a subtle change in our previous logical interpretation of dependent types: Here the dependent function taking any point to a path to the *center of contraction* has to be interpreted as being continuous in some way, otherwise the notion we would be describing would be that of a space being path connected. We will later modify this definition so that continuity is not guaranteed using propositions.

3.4.6 Definition.

- Let $f, g : A \to B$ be two functions. A ***homotopy*** between f and g is a dependent function of type
$$(f \sim g) :\equiv \prod_{x:A} (f(x) = g(x))$$
- A function $f : A \to B$ is an ***equivalence*** if there is an element in the type (actually proposition!)
$$\mathsf{isequiv}(f) :\equiv \Big(\sum_{g:B\to A} (f \circ g \sim \mathsf{id}_B)\Big) \times \Big(\sum_{h:B\to A} (h \circ f \sim \mathsf{id}_B)\Big)$$

3.4.7 Definition. Let $f : A \to B$ be a function and $b : B$. We define the ***fiber*** of f at b to be the type
$$\mathsf{fib}_f(b) :\equiv \sum_{x:A} (f(x) = b)$$

3.4.8 Theorem. *A function $f : A \to B$ is an equivalence if and only if it has contractible fibers, i.e., if there is an element of the type*
$$\prod_{b:B} \mathsf{isContr}(\mathsf{fib}_f(b))$$

Type families as fibrations

Let $C : A \to \mathcal{U}$ be a type family. Given that we have given a homotopical interpretation to paths, we should be aple to interpret $\sum$ and $\prod$ types in a similar way. We begin with $\sum$-types: Recall that we can define the first projection $\mathrm{pr}_1 : \sum_{x:A} C(x) \to A$. In the following results, we will provide motivation for thinking of $\sum_{x:A} C(x)$ as the ***total space*** of the fibration $\mathrm{pr}_1 : \sum_{x:A} C(x) \to A$, with $C(x)$ being the ***fiber*** over x.

For now, recall the notion of a *fiber bundle* of topological spaces, a particular kind of Serre fibration $q : E \to B$ where the fibers $p^{-1}(b)$ locally look like products $E \times F$ such that q locally looks like the projection.

Moving to $\prod$-types, consider a dependent function $f : \prod_{x:A} C(x)$. For $x : A$, observe that we can define a function
$$\tilde{f} :\equiv \lambda x.(x, f(x)) : \prod_{x:A} \Big(\sum_{y:A} C(y)\Big)$$
and now $\mathrm{pr}_1(\tilde{f}(x)) \equiv x : A$. We thus think of $\prod_{x:A} C(x)$ as the type of ***sections*** of the fibration $\sum_{x:A} C(x) \xrightarrow{\mathrm{pr}_1} A$.

This discussion can be packaged into this syntactically highly informal, but semantically accurate pullback diagram (assuming $a : A$ and $f : \prod_{x:A} C(x)$):

$$\begin{tikzcd}
C(a) \arrow[r] \arrow[d] & \sum_{x:A} C(x) \arrow[d, "\mathrm{pr}_1"'] \\
\mathbf{1} \arrow[r, "a"] & A \arrow[u, bend right, dotted, "\tilde{f}"']
\end{tikzcd}$$

We now state some more results that describe the behavior of $\sum$-types; we start with a characterization of paths in $\sum$-types that states that they are made up precisely of pairs of paths in the base and compatible paths in the fibers.

3.4.9 Theorem ([Uni13, Theorem 2.7.2]). *Let $C : A \to \mathcal{U}$ be a type family and $u, v : \sum_{x:A} C(x)$. There is an equivalence*

$$(u =_{\sum_{x:A} C(x)} v) \simeq \sum_{(p:\mathrm{pr}_1(u)=_A \mathrm{pr}_1(v))} p_*(\mathrm{pr}_2(u)) =_{C(\mathrm{pr}_1(v))} \mathrm{pr}_2(v)$$

Next, we present the *path lifting property* for $\sum$-types. This can be thought of as the defining property of fibrations:

3.4.10 Lemma ([Uni13, Lemma 2.3.2]). *Let $C : A \to \mathcal{U}$ be a type family, $a : A$ and $z : C(a)$. For any $p : a =_A b$, we have a lifting function*

$$\mathsf{lift}(z, p) : (a, z) =_{\sum_{x:A} C(x)} (b, p_*(a))$$

such that $\mathrm{pr}_1(\mathsf{lift}(z, p)) =_{(a=b)} p$.

3.5. The Univalence Axiom

The notion of univalence comes from the observation that mathematical structures that are equivalent in some relevant sense tend to be identified. In classical mathematics, this is an informal process; one has to prove that the corresponding equivalece preserves any properties we care about. In type theory, univalence allows us to make this identification using identity types.

3.5.1 Construction. Let $\mathcal{U}$ be a universe. For any two types $A, B : \mathcal{U}$ we can define a map

$$\mathsf{idtoequiv} : (A =_{\mathcal{U}} B) \to (A \simeq_{\mathcal{U}} B)$$

given by sending refl_A to id_A and applying transport and path induction.

3.5.2 Definition. A universe $\mathcal{U}$ is ***univalent*** if, for all types $A, B : \mathcal{U}$, the function $\mathsf{idtoequiv}$ is an equivalence.

One very powerful result implied by univalence is the following type-theoretic analogue of the Grothendieck construction:

3.5.3 Theorem. *For any type $A : \mathcal{U}$ where $\mathcal{U}$ is a univalent universe, the map*

$$\Big(\sum_{X:\mathcal{U}} (X \to A)\Big) \to (A \to \mathcal{U})$$
$$(X, f) \mapsto \mathsf{fib}_f$$

is an equivalence.

Recall that we can see $A \to \mathcal{U}$ as the type of type families indexed by A. Then this result is saying that the full homotopical structure of such families is encoded in the type of maps into A.

3.5.4 Theorem ([Uni13, Section 4.9])**.** *Univalence implies the function extensionality axiom: In a univalent universe $\mathcal{U}$, for any type family $C : A \to \mathcal{U}$ and any $f, g : \prod_{x:A} C(x)$, the function*

$$\mathsf{htpy{-}eq}_{f,g} : (f = g) \to (f \sim g)$$

defined by path induction is an equivalence.

3.6. Higher inductive types, truncations and more homotopy theory

Definitions of higher inductive types are a general schema of inductive definitions, where we require the existence of nontrivial paths in some higher level of identity types. For the purposes of this section, we will give an overview of the case of the circle.

3.6.1 Example ([Uni13, Section 6.4])**.** The ***circle*** $\mathbb{S}^1$ is the type generated by:

- A point $\mathsf{base} : \mathbb{S}^1$.
- A path $\mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}$.

together with the induction principle that constructs maps out of the circle: For a family $P : \mathbb{S}^1 \to \mathcal{U}$, a point $b : P(\mathsf{base})$ and a ***dependent path*** $l : b =^P_{\mathsf{loop}} b$ over loop, there is a dependent function $f : \prod_{x:\mathbb{S}^1} P(x)$ with $f(base) \equiv b$ and $\mathsf{apd}_f(\mathsf{loop}) = l$.

Using the loop space construction that we will define below, we can also define higher-dimensional spheres.

We will now discuss operations that "forget the higher structure" of types. This serves both a logical and a homotopy-theoretic perspective: In homotopy theory, recall the Postnikov truncations of a space, which kill its homotopy groups at some level and above. Focusing on the lowest levels, we can only remember if a type is inhabited or not (propositions), or remove all higher paths (sets). In the next section, we will use propositional truncations to do classical logic in homotopy type theory.

3.6.2 Definition. Let $n \geq -2$. We say that a type X is an ***n-type*** if the type is-n-type(X) is inhabited, defined recursively as

$$\begin{aligned}\text{is-}(-2)\text{-type}(X) &:\equiv \mathsf{isContr}(X)\\ \text{is-}(n+1)\text{-type}(X) &:\equiv \prod_{x,y:X} \text{is-}n\text{-type}(x =_X y)\end{aligned}$$

We say that a map $f : X \to Y$ ***n-truncated*** if all its fibers are n-types.

3.6.3 Definition. In particular, we call (-1)-types ***propositions***, and 0-types ***sets***.

We mention the following important results about sets:

3.6.4 Theorem ([Uni13, Theorem 7.2.1]). *A type X is a set if and only if it satisfies **Axiom K**: For any $a : X$ and any $p : a =_X a$, we have path $p =_{(a=_X a)} \mathsf{refl}_a$*

3.6.5 Theorem (Hedberg's theorem). *If X has **decidable equality** (meaning that all identity types are decidable), then X is a set.*

3.6.6 Theorem ([Uni13, Theorem 7.1.7]). *Every n-type is an $n + 1$-type.*

Proof. Let x be an n-type. Induction on n:

$n = -2$. X is contractible, so assume $x_0 : X$ is the center of contraction. We want to show that every identity type $x =_X y$ is contractible. By contractibility of X, we have paths $\mathrm{contr}_x : x =_X x_0$ and $\mathrm{contr}_y : y =_X x_0$. Choose the path $\mathrm{concat}(\mathrm{contr}_x, \mathrm{contr}_x^{-1})$ to be the center of contraction for $x =_X y$ and use path induction and the fact that there is a path in $\mathrm{concat}(\mathrm{contr}_x, \mathrm{contr}_x^{-1}) =_{(x=_X x)} \mathsf{refl}_x$.

$n = m + 1$. We want to show that every identity type $x =_X y$ is an m-type, which is true by the induction hypothesis.

□

3.6.7 Theorem ([Uni13, 7.1.8, 7.1.9]). *Let $C : A \to \mathcal{U}$ be a type family and assume that all fibers $B(a)$ are n-types. Then:*

1. *$\prod_{x:A} C(x)$ is an n-type.*
2. *If A is also an n-type, $\sum_{x:A} C(x)$ is an n-type.*

3.6.8 Theorem ([Rij22, Theorem 12.4.7]). *A map $f : X \to Y$ is $(n+1)$-truncated if and only if for every $x, y : X$, the map $\mathsf{ap}_f : (x =_X y) \to (f(x) =_Y f(y))$ is n-truncated.*

3.6.9 Definition (Universal property of the truncation). Let X be a type. A map $f : X \to Y$ satisfies the universal property of the ***n-truncation*** if, for any n-type Z, the precomposition map $- \circ f : (Y \to Z) \to (X \to Z)$ is an equivalence.

3.6.10 Theorem (Existence of n-truncations, [Uni13, Section 7.3]). *There is an n-truncation function $|| - ||_n$ for every n.*

In particular, we have the *propositional truncation* $\|-\|_{-1}$ and the *set truncation* $\|-\|_0$.

3.6.11 Construction. We can use truncations as well as the higher groupoidal structure of identity types to define homotopy groups of types: For a type A and a chosen basepoint $a : A$, we can define the loop space

$$\Omega(A, a) :\equiv a =_A a$$

together with the chosen basepoint refl_a, and iteratively define higher loop spaces $\Omega^{n+1}(A, a) :\equiv \Omega(\Omega^n(A, a))$. Now set

$$\pi_n(A, a) :\equiv \|\Omega^n(A, a)\|_0$$

3.6.12 Remark. Since this definition of homotopy groups uses this looping construction as its core, let us do a quick sanity check and convince ourselves that it captures the same information as the usual constructions in spaces or Kan complexes.

For a Kan complex X with a fixed vertex x_0, define its *path space* to be the pullback

$$PX :\equiv X_1 \times_X \Delta^0$$

i.e., the fiber of $X_1 \xrightarrow{d_0} X$ along x_0. We can then define the *loop space*

$$\begin{tikzcd}
\Omega X \arrow[r] \arrow[dd] \arrow[dr, phantom, "\lrcorner", very near start] & PX \arrow[dd] \arrow[dr] & \\
 & & X_1 \arrow[dl, "d_1"] \\
\Delta^0 \arrow[r, "x_0"] & X &
\end{tikzcd}$$

To show that this definition is equivalent to the one we gave in chapter 2, we can iteratively use the fact that the suspension-loop adjunction carries over to Kan complexes. Then, since $(\Delta^{n+1}, \partial\Delta^{n+1}) \simeq (\Sigma\Delta^n, \Sigma\partial\Delta^n)$, taking path components results in the corresponding homotopy group.

Note that some sources even define homotopy groups of Kan complexes this way, see for example [Cis19, 3.8].

3.7. Logical aspects, part II: Propositions and comparisons with ZFC

Now that we have the propositional truncation of a type, we can make a precise analogy to classical logic. Applying $\|-\|_{-1}$ to a type keeps track of whether the type is inhabited or not. We then translate first-order logic as follows: Assume that P and Q are propositions and $R : A \to \mathcal{U}$ is a family of propositions over a type A. We define the correspondence

$P \vee Q$	$\|P + Q\|_{-1}$
$\exists(x : A)R(x)$	$\sum_{x:A} \|R(x)\|_{-1}$
$\forall(x : A)R(x)$	$\prod_{x:A} R(x)$

Note that $P \times Q$ and $\prod_{x:A} R(x)$ are already propositions, so no modifications have to be made!

We refer to $\sum_{x:A} ||R(x)||_{-1}$ as "*there merely exists* $x : A$ such that $R(x)$".

3.7.1 Remark. The statement "there merely exists (...)" can be interpreted as "there exists, but not in a continuous/uniform manner". For example, recall the definition of a type being contractible. If we change the the analogous statement $\sigma_{a_0:A} \prod_{x:A} ||(a_0 =_A x)||_{-1}$, this statement now means that A is *path connected*.

In that interpretation of logic, we may assume the axiom of choice or the law of excluded middle for propositions. However, in the full type theory, the law of excluded middle is false [Uni13, Theorem 3.2.2], but a constructive version of the axiom of choice is true, as, the choice function is given to us by default. We will see that "axiom" (actually a theorem) in a different setting in Chapter 5.

Assuming the full type theory we have defined so far, types with higher structure have to exist:

3.7.2 Proposition. *Assuming $\mathcal{U}$ is univalent, it is not a set.*

Proof sketch. We can define two different self-equivalences of the type of bool: the identity and the map $\lambda \mathsf{true}.\mathsf{false}, \lambda \mathsf{false}.\mathsf{true}$. If $\mathcal{U}$ is a set, then univalence implies that there is a path between these two equivalences, which means that $\mathsf{false} = \mathsf{true}$, contradicting the basic theory of coproducts. ☐

4. Interlude: On Univalence and Rezk Completeness

In this chapter, we present a survey of examples and relations of univalence and the condition of an object being Rezk-complete. The goal is to highlight how these notions provide the means for a coherent theory of various forms of equivalence that, by adding extra structure, can be made to behave as an identification of two elements.

4.1. Flagged categories and 2-categorical equivalences

Recall that in 1-category theory, we are usually not working up to strict isomorphism of categories, but up to the weaker notion of an *equivalence of categories*: A functor together with an inverse up to natural isomorphism. We can make the difference precise using 2-categories, as a special case of work in [AF18]:

4.1.1 Definition. The $(2,1)$-category $\mathcal{C}\mathrm{at}$ is defined by the following data:

- Objects are categories.
- Morphisms are functors.
- 2-morphisms are natural isomorphisms.

4.1.2 Definition. A ***flagged*** $(2,1)$***-category*** is an essentially surjective functor $G \to C$ of categories where the domain is a groupoid. They assemble naturally into a $(2,1)$-category $f\mathcal{C}at$ with objects flagged categories, morphisms functors that make the evident square commute and 2-morphisms natural isomorphisms that fit into a commutative diagram

$$\begin{tikzcd}
G_1 \arrow[r, bend left, "a"{name=A}] \arrow[r, bend right, "b"'{name=B}] \arrow[d, "F_1"'] & G_2 \arrow[d, "F_2"] \\
\mathcal{C}_1 \arrow[r, bend left, "k"{name=K}] \arrow[r, bend right, "l"'{name=L}] & \mathcal{C}_2
\arrow[Rightarrow, from=A, to=B, "\mu"'] \arrow[Rightarrow, from=K, to=L, "\nu"]
\end{tikzcd}$$

4.1.3 Lemma. *Let* $F_1 : G_1 \xrightarrow{F_1} \mathcal{C}_1, F_2 : G_2 \xrightarrow{F_2} \mathcal{C}_2$ *be two objects in* $f\mathcal{C}at$*, such that* G_1, G_2 *are discrete and* F_1*,* F_2 *are surjective on objects, then the groupoid* $\mathrm{Hom}(F_1, F_2)$ *is discrete.*

Proof. Let (a,k), (b,l) and (μ,ν) be as in the diagram above. where μ and ν are natural isomorphisms. Since there are no non-identity morphisms in G_2, we get that $a = b$ and μ is the identity. Now let g be an object of G_1. Since $a = b$, the diagram now says that $id_{F_2a(g)} = id_{lF_1(g)} = \nu_{F_1(g)}$. Thus $k(g) = l(g)$. Since F_1 is surjective, we get our result. ☐

4.1.4 Proposition. *There are fully faithful* $(2,1)$*-embeddings*

$$\begin{aligned} \mathrm{CAT} &\xhookrightarrow{i_S} f\mathcal{C}at & \qquad \mathcal{C}\mathrm{at} &\xhookrightarrow{i_W} f\mathcal{C}at \\ \mathcal{C} &\mapsto (\mathrm{Obj}(\mathcal{C}) \to \mathcal{C}) & \qquad \mathcal{C} &\mapsto (\mathcal{C}^{\simeq} \to \mathcal{C}) \end{aligned}$$

where CAT *is the* $(1,1)$*-category of categories, taking a category to the inclusion of the set of objects, and the* $(2,1)$*-category of categories, taking a category to its groupoid core inclusion.*

Proof sketch. In the case of the first embedding, we use the fact that hom-groupoids are discrete by the previous lemma.

For the second embedding, let $F_1, F_2 : \mathcal{C} \to \mathcal{D}$, we have to show that $\mathrm{Nat}(F_1, F_2)$ is in bijection (induced by i_W) with the set of morphisms between $(F_1^{\simeq}, F_1)$ and $(F_2^{\simeq}, F_2)$. This is equivalent to saying that for a natural isomorphism ν, the only natural isomorphism that restricts to ν between groupoid cores is ν itself, which is true since ν consists of isomorphisms. For a commmutative square

$$\begin{tikzcd} \mathcal{C}^{\simeq} \arrow[r, "a"] \arrow[d] & \mathcal{D}^{\simeq} \arrow[d] \\ \mathcal{C} \arrow[r, "b"'] & \mathcal{D} \end{tikzcd}$$

we immediately get that a is the restriction $b^{\simeq}$, so this is precisely the image of $\mathcal{C} \xrightarrow{b} \mathcal{D}$. □

With these embeddings into $f\mathcal{C}at$, we now have a unified framework that contains both strict categories (the image of CAT, where equivalences are isomorphisms of categories) and "univalent" categories (the image of $\mathcal{C}$at, where equivalences are equivalences of categories).

In that case, the *Rezk completion* functor is the functor $i_S(\mathrm{CAT}) \to i_W(\mathcal{C}\mathrm{at})$ induced by the inclusion $\mathrm{Obj}(\mathcal{C}) \hookrightarrow \mathcal{C}^{\simeq}$.

4.2. Warmup: univalent 1-categories in homotopy type theory

We outline the approach to do 1-category theory in homotopy type theory, as laid out in [AKS15]. Here, we can define a 1-category without any extra theory, as we require types of morphisms to have no higher structure. Moreover, a "local univalence"/Rezk completeness condition will again play a role, with the property that, as with Segal spaces, there is a completion functor turning a precategory into a univalent 1-category.

4.2.1 Definition ([AKS15, Definition 3.1]). A ***precategory*** A consists of:

1. A type of ***objects*** A_0. We write $a : A$ for $A : A_0$.
2. For each $a, b : A$, a *set* $\hom_A(a,b)$ of ***morphisms*** or ***arrows***.
3. For each $a : A$, an ***identity morphism*** $1_a : \hom_A(a,a)$.
4. For each $a, b, c : A$ and $f : \hom_A(a,b)$, $g : \hom_A(b,c)$, a ***composition function***

$$- \circ - : \hom_A(a,b) \to \hom_A(b,c) \to \hom_A(a,c)$$

5. Witnesses for the properties of composition:
    - For each $a, b : A$, functions of type

$$\prod_{(f:\mathrm{hom}_A(a,b))} (f = f \circ 1_a) \quad \text{and} \quad \prod_{(f:\mathrm{hom}_A(a,b))} (f = 1_b \circ f)$$

    - For each $a, b, c, d : A$ and $f : \mathrm{hom}_A(a, b)$, $g : \mathrm{hom}_A(b, c)$, $h : \mathrm{hom}_A(c, d)$, a witness for associativity: $h \circ (g \circ f) = (h \circ g) \circ f$.

We can define *functors* between precategories and *natural transformations* in the same way. For precategories A, B, we define by B^A the correspoding category of functors from A to B.

4.2.2 Definition. 1. For $f : \mathrm{hom}_A(x, y)$ define the type

$$\mathsf{isiso}(f) := \Big(\sum_{g:\mathrm{hom}_A(y,x)} g \circ f = 1_x \Big) \times \Big(\sum_{h:\mathrm{hom}_A(y,x)} f \circ h = 1_y \Big)$$

f is an ***isomorphism*** if $\mathsf{isiso}(f)$ is inhabited.

2. For fixed $x, y : A$, denote by

$$(x \cong_A y) := \sum_{f:\mathrm{hom}_A(x,y)} \mathsf{isiso}(f)$$

the type of all isomorphisms between x and y.

We now express Rezk completeness: Here, we want to unify identities and isomorphisms, as we just defined them.

4.2.3 Construction. Let A be a precategory and $a, b : A$. By path induction, we can construct a map

$$\mathsf{idtoiso}_{a,b} : (a = b) \to (a \cong b)\,, \quad \mathsf{idtoiso}(\mathsf{refl}_a) :\equiv 1_a$$

4.2.4 Definition. A precategory A is a ***category*** if, for all $a, b : A$, $\mathsf{idtoiso}_{a,b}$ is an equivalence.

For an example of why the category property is needed, we can prove that for categories, all notions of equivalence that we can define coincide (see [AKS15, Section 6]).

4.2.5 Proposition ([AKS15, Theorem 4.5]). *If B is a category and A is a precategory, then B^A is a category.*

4.2.6 Example (The category of sets). Fix a universe $\mathcal{U}$. The type $\mathsf{Set}_{\mathcal{U}}$ of sets assembles into a precategory with $\mathrm{hom}_{\mathsf{Set}_{\mathcal{U}}}(A, B) :\equiv (A \to B)$. Then, if $\mathcal{U}$ is univalent, $\mathsf{Set}_{\mathcal{U}}$ is a category.

Now that we have a category of sets, we can define a Yoneda embedding. Fix a precategory A with $A_0 : \mathcal{U}$. By [AKS15, Lemma 7.3], the hom-functor $\hom_A(-,-) : A \times A^{op} \to \mathsf{Set}_{\mathcal{U}}$ induces the yoneda embedding

$$\mathcal{y} : A \to (\mathsf{Set}_{\mathcal{U}})^{A^{op}}$$

which, by the Yoneda lemma [AKS15, Lemma 7.4], is fully faithful. We finally arrive at the Rezk completion for precategories:

4.2.7 Theorem (The Rezk completion for precategories [AKS15, Theorem 8.5])**.** *Let A be a precategory with $A_0 : \mathcal{U}$. Let*

$$\hat{A} :\equiv \left\{F \in (\mathsf{Set}_{\mathcal{U}})^{A^{op}} | \exists (a : A). \mathcal{y}(A) \cong F\right\}$$

be the full subcategory of representables in $(\mathsf{Set}_{\mathcal{U}})^{A^{op}}$. Then $\hat{A}$ is a category and the yoneda embedding $A \to \hat{A}$ is a weak equivalence.

4.2.8 Remark. This construction works only when assuming enough univalent universes, as $\hat{A}$ lives in a higher universe than $\mathcal{U}$. For that reason, no such construction is provided in simplicial type theory (yet!), even though we have a completion functor for Segal spaces [Rez01, Section 14]. In the context of homotopy type theory, a concrete construction that takes care of the universe issue is provided in [Uni13, Section 9.9].

Unlike our discussion so far, univalence and completeness can be related in precise ways. For such higher categorical interpretations, we point to [Ste23] (in the setting of type-theoretic fibration categories, where type theories can be interpreted) and [Ras21].

5. Simplicial Type Theory

5.1. Introduction

To say that an extension of homotopy type theory has a model in Segal spaces, we first need to make sure that we can interpret HoTT itself in such a model. We therefore start with this result:

5.1.1 Theorem ([Shu15, Theorem 6.4]). *The category $s\mathcal{S}$ of simplicial spaces with the Reedy model structure (actually, every category of the form* $\mathrm{Fun}(\mathcal{C}^{op}, s\mathcal{S}\mathrm{et})$ *where $\mathcal{C}$ is an elegant Reedy category) supports a model of intensional type theory with dependent sums and products, identity types, and as many univalent universes as inaccessible cardinals (greater than $|\mathcal{C}|$).*

Interpreting types as fibrant objects in the Reedy model structure allows us to simplify the Segal condition enough so that it can be expressed in type theory. By combinatorial arguments (based on work of Joyal [Joy]), the Segal condition can be reduced as follows:

5.1.2 Theorem ([RS17, Theorem A.21]). *A Reedy fibrant simplicial space W is Segal if and only if the map*

$$W^{F(2)} \to W^{L(2)_1}$$

is a trivial Reedy fibration.

We can now fully define the type theory we will be working with, where it is possible to express the property of a type being a pre-∞-category (Segal) and an ∞-category (Rezk). Benefitting from the fact that types are interpreted as fibrant objects in the first place, for the Segal condition we need only check that the analogous map is an equivalence. Having guaranteed uniqueness of composition up to homotopy for Segal types, the Rezk condition can then be defined in the same way as in the previous "warmup" chapter (4.2.4).

The main foundational step is to define a theory of simplices that can extract types of objcets, morphisms, etc., as in set-based simplicial homotopy theory. The process goes roughly as follows:

1. Start with a simple logical layer, aiming to define simplices by "geometric" relations.
2. Provide axioms for a ***directed interval type* 2**, maps out of which represent directed arrows (the analogue of Δ^1).
3. Define simplices and their horns and boundaries by imposing conditions on cubes of the form $\mathbb{2} \times \cdots \times \mathbb{2}$.

4. Introduce ***extension types***, which represent lifting problems. Then, the horn-fillings can be expressed as such lifting problems.

Of course, the hom-types that simplicial type theory adds will themselves be types! We have, again, an analogue in simplicial spaces:

5.1.3 Definition. Let W be a simplicial space and $x, y : F(0) \to W$. The ***mapping simplicial space*** $\mathrm{Hom}_W(x, y)$ is defined as the pullback of simplicial spaces

$$\begin{tikzcd}
\mathrm{Hom}_W(x,y) \arrow[r] \arrow[d] & W^{F(1)} \arrow[d] \\
F(0) \arrow[r, "{(x,y)}"] & W \times W
\end{tikzcd}$$

The mapping Kan complex $\mathrm{Map}_W(x, y)$ in a Segal space is the 0-th level of $\mathrm{Hom}_W(x, y)$, but the Segal condition guarantees that it already contains all the homotopical information.

5.1.4 Remark. Rezk types come in handy in various places where full control over equivalences is needed, in particular when formalizing adjunctions ([RS17, Chapter 11]) and limits ([Mar23]) or when doing more general work with fibrations ([BW23]), but a lot of work can be done with Segal types alone (for example, the ∞-categorical Yoneda lemma). In this thesis, we have focused on properties of Segal types.

Statements from now on are accompanied by formalization in the Rzk proof assistant, developed specifically for simplicial type theory by Nikolai Kudasov. The main feature of Rzk is its *tope solver*, allowing for automatic checking of the relevant restrictions when defining shapes and terms in extension types.

5.2. Axioms for shape construction

The logical basis for defining simplices as shapes consists of two layers: cubes and topes. Cubes are generated by the directed interval $\mathbb{2}$ and topes impose relations on cubes, and thus a shape will be defined as a cube (which the shape embeds into) together with a tope.

Geometrically, think of the topological n-simplex

$$|\Delta^n| := \left\{(x_0, \dots, x_n) \in (\mathbb{R}_{\geq 0})^{n+1} \,\middle|\, \sum_{i=0}^{n} x_i = 1\right\}$$

embedding into the cube $[0, 1]^{n+1}$. $|\Delta^n|$ is homeomorphic to the subspace

$$\left\{(x_1, \dots, x_n) \in [0, 1]^n \,\middle|\, x_1 \leq \dots \leq x_n\right\}$$

which is now generated only by inequality relations! Imposing further conditions, we can define boundaries and horns, and this is exactly how we will proceed in simplicial type theory.

Cubes

We begin with a simple theory of cubes: To define these, we really only need a theory of finite products, since cubes are all made up of products of other cubes. We first introduce the primitive generators of the cube layer, **1** and $\mathbb{2}$

$$\frac{\Gamma\ \mathsf{ctx}}{\mathbf{1}\ \mathsf{cube}} \qquad\qquad \frac{\Gamma\ \mathsf{ctx}}{\star : \mathbf{1}}$$

$$\frac{\Gamma\ \mathsf{ctx}}{\mathbb{2}\ \mathsf{cube}} \qquad\qquad \frac{\Gamma\ \mathsf{ctx}}{0 : \mathbb{2}} \qquad\qquad \frac{\Gamma\ \mathsf{ctx}}{1 : \mathbb{2}}$$

and we then present the theory of finite products, with a rule for constructing terms and two projections:

$$\frac{(t : I) \in \Gamma}{\Gamma \vdash t : I} \quad \frac{I\ \mathsf{cube} \quad J\ \mathsf{cube}}{I \times J\ \mathsf{cube}} \quad \frac{\Gamma \vdash s : I \quad \Gamma \vdash t : J}{\Gamma \vdash \langle s, t\rangle : I \times J} \quad \frac{\Gamma \vdash r : I \times J}{\Gamma \vdash \pi_1(r) : I} \quad \frac{\Gamma \vdash r : I \times J}{\Gamma \vdash \pi_2(r) : J}$$

5.2.1 Remark. Type-theoretically, there is nothing essential about cubes and shapes being based on the directed interval $\mathbb{2}$. In fact, in [RS17, Chapter 2], Riehl and Shulman provide the general framework for a type theory with shapes before introducing the directed interval or its theory.

Topes

The tope layer consists of axioms for logical formulas. In particular, it is an *intuitionistic logic* consisting of:

- $\top$ and $\bot$ symbols, with their usual behavior.
- For topes ϕ, ψ, intuitionistic disjunction $\phi \vee \psi$ and conjunction $\phi \wedge \psi$.
- For I cube and $s, t : I$, a strict ***equality tope*** $s \equiv t$.
- Euality respects substitution, cube products and projections: e.g. $\dfrac{r : I \times J}{r \equiv \langle \pi_1(r), \pi_2(r)\rangle}$

Then, we have to describe the function of the generating cubes, i.e., the one-point cube **1** and the directed interval $\mathbb{2}$. For **1**, there is only one strict uniqueness rule

$$\frac{\Gamma \vdash r : \mathbf{1}}{\Gamma \vdash r \equiv \star}$$

In order to axiomatize $\mathbb{2}$ as a directed interval, it comes with an ***inequality tope***

$$\frac{x, y : \mathbb{2}}{(x \leq y)\ \mathsf{tope}}$$

which comes with:

- Reflexivity, symmetry (with respect to the equality tope) and transitivity axioms.
- Interaction of terms with the two specified endpoints $0 : \mathbb{2}$ and $1 : \mathbb{2}$.

$$\frac{\Gamma \vdash x : \mathbb{2}}{\Gamma \vdash 0 \leq x} \quad \frac{\Gamma \vdash x : \mathbb{2}}{\Gamma \vdash x \leq 1} \quad \frac{}{0 \equiv 1 \vdash \bot}$$

Shapes

We can now create shapes as topes together with topes. The introduction rule says exactly that:

$$\frac{I \text{ cube} \quad t : I \vdash \phi \text{ tope}}{\{t : I \mid \phi\} \text{ shape}}$$

The relations above are now enough to define simplices.

5.2.2 Definition. We define the ***n-simplex***

$$\Delta^n := \{\langle t_1, \dots, t_n \rangle : \mathbb{2}^n \mid t_n \leq \dots \leq t_1\}$$

5.2.3 Example. Noticing that the condition on the right always holds for the first two simplices, we can simplify to $\Delta^0 = \{t : \mathbf{1} \mid \top\}$ and $\Delta^1 = \{t : \mathbb{2} \mid \top\}$, i.e., the point and the interval.

Adding more conditions to our topes, we can now define basic boundaries and horns as subshapes of our simplices:

- $\partial\Delta^1 := \{t : \mathbb{2} \mid t \equiv 0 \vee t \equiv 1\}$ ("endpoints of $\mathbb{2}$")
- $\partial\Delta^2 := \{\langle t_1, t_2 \rangle : \mathbb{2} \times \mathbb{2} \mid (0 \equiv t_2) \vee (t_1 \equiv t_2) \vee (t_1 \equiv 1)\}$ (boundary defined by two sides and the diagonal)
- $\Lambda^2_1 := \{\langle t_1, t_2 \rangle : \mathbb{2} \times \mathbb{2} \mid (0 \equiv t_2) \vee (t_1 \equiv 1)\}$

5.2.4 Remark. Note that we did not define horns and boundaries in general. However, in practice we will only need low-dimensional simplices and boundaries, usually of dimension at most 3.

Mapping out of shapes by $\vee$-recursion. For shapes $\{t : I \mid \phi\}$, we introduce an additional dependent product type

$$\prod_{t:I|\phi} A(t)$$

Such a type will have its formation rules, which are recursive on $\vee$:

To construct a dependent function $f : \prod_{t:I|\phi\vee\psi} A(t)$, all we have to do is to provide functions out of $\{t : I \mid \phi\}$ and $\{t : I \mid \psi\}$ which agree when $\phi \wedge \psi$ holds.

5.2.5 Remark. We do not yet provide a theory of universes. Rzk works with one universe level (and no higher inductive types), but a lot of the basic theory can already be expressed this way.

5.3. Extension types

As is the case with all types we used before constructing shapes, one can define (dependent) function types where the base type is a shape. Since the goal here is to express equivalents of filling conditions, a theory of *liftings* is needed. Dependent function types out of shapes therefore come with the more general notion of an extension type along a subspace inclusion.

5.3.1 Notation. For a given cube I, we write $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$ if $t : I \mid \phi \vdash \psi$.

Then, for a shape inclusion $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$, we can introduce the extension type

$$\frac{\begin{array}{c}\{t : I \mid \phi\}\ \mathsf{shape} \quad \{t : I \mid \psi\}\ \mathsf{shape} \quad t : I \mid \phi \vdash \psi \\ \Xi|\Phi \vdash \Gamma\ \mathsf{ctx} \quad \Xi, t : I \mid \Phi, \phi \mid \Gamma \vdash A\ \mathsf{type} \quad \Xi, t : I \mid \Phi, \phi \mid \Gamma \vdash a : A\end{array}}{\Xi|\Phi|\Gamma \vdash \Big\langle \prod_{t:I|\psi} A(t)|^{\phi}_{a} \Big\rangle\ \mathsf{type}}$$

We think of an extension type $\langle\prod_{t:I|\psi} A|^{\phi}_{a}\rangle$ as the type of dependent functions that, when restricted to the subshape $\{t : I \mid \phi\}$ (i.e., when ϕ holds), map to a. Informally (and for a constant family A) what we want is a type of lifting diagrams

$$\begin{tikzcd}
\{t : I \mid \phi\} \arrow[r] \arrow[d, hook] & A \\
\{t : I \mid \psi\} \arrow[ur, dotted] &
\end{tikzcd}$$

The rest of the type-forming rules for extension types are the same as the usual dependent product types, adding the restriction condition (judgmentally) for elimination rules. In formal terms:

$$\frac{\begin{array}{c}\{t : I \mid \phi\}\ \mathsf{shape} \quad \{t : I \mid \psi\}\ \mathsf{shape} \quad t : I \mid \phi \vdash \psi \\ \Xi \mid \Phi \mid \Gamma \vdash f : \langle\prod_{t:I|\psi} A(t)|^{\phi}_{a}\rangle \quad \Xi \vdash s : I \quad \Xi \mid \Phi \vdash \phi(s)\end{array}}{\Xi \mid \Phi \mid \Gamma \vdash f(s) \equiv a[s/t]}$$

Semantically, under a chosen interpretation, extension types fit into a pullback diagram as below:

$$\begin{tikzcd}
\langle\prod_{t:\Psi} A(t)|^{\Phi}_{a}\rangle \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\lrcorner", very near start] & A^{\Psi} \arrow[d, two heads] \\
\Gamma \arrow[r, "\langle \tilde{a}\, , id_{\Gamma\times\Psi}\rangle"] & A^{\Phi} \times_{(\Gamma\times\Psi)^{\Phi}} (\Gamma \times \Psi)^{\Psi}
\end{tikzcd}$$

Here, the bottom map says that we have $a : \prod_{\{t:I \mid \phi\}} A$ in context Γ, which we want to extend to A^{Ψ}, or $\prod_{\{t:I \mid \psi\}} A$. For more on the semantics of shapes and extension types, see [RS17, Theorem A.16] and [Wei22b].

5.3.2 Remark (Extension types up to homotopy, [BW23, Section 2.4]). Since we have worked with identity types so far, it is worthwhile to ask why the equality in the restriction part of an extension type is judgmental, and if any homotopical information is lost. If one assumes that shapes are themselves types (otherwise the statement does not make sense), Buchholtz and Weinberger prove that the different ways to define such types are, in fact, equivalent.

Recall that in 3.4, we interpreted dependent functions as sections of the first projection $\sum_{x:A} C(x) \to A$. In particular, $f : \prod_{x:A} C(x)$ takes a point $a : A$ in the base to a point $f(a) : C(a)$ in the fiber over a. If $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$ is a shape inclusion and $f : \left\langle \prod_{t:I|\psi} A(t)|^{\phi}_{a} \right\rangle$, we can define a function of type

$$\prod_{t:I|\phi} (a(t) = f(t))$$

with the witness being simply refl.

We can now define the ***homotopy extension type*** (not to be confused with the homotopy extension property later on) as the $\sum$-type with the restriction condition appearing in the fibers, which are intensional identity types. Then, we can prove the following equivalence between the judgmental and homotopy extension type:

$$\left\langle \prod_{t:I|\psi} A(t)|^{\phi}_{a} \right\rangle \simeq \sum_{(f:\prod_{t:I|\psi} A(t))} \prod_{(t:I|\phi)} (a(t) = f(t))$$

The definition of homotopy extension types and this theorem have been formalized in Rzk by Tashi Walde (🔗).

Next, we restate results that hold for ordinary $\prod$-types in the language of extension types.

5.3.3 Theorem (“Axiom” of choice, [RS17, Theorem 4.3]). *For a shape inclusion* $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$*, a family* $X : \{t : I \mid \psi\} \to \mathcal{U}$*,* $Y : \prod_{t:I|\psi}(X \to \mathcal{U})$*, and dependent functions* $a : \prod_{t:I|\phi} X(t)$*,* $b : \prod_{t:I|\phi} Y(t, a(t))$*, there is an equivalence*

$$\left\langle \prod_{t:I|\psi} \Big(\sum_{x:X(t)} Y(t,x) \Big)\Big|^{\phi}_{\lambda t.(a(t),b(t))} \right\rangle \simeq \sum_{f:\langle \prod_{t:I|\psi} X(t)|^{\phi}_{a} \rangle} \left\langle \prod_{t:I|\psi} Y(t, f(t))|^{\phi}_{b} \right\rangle$$

Proof. We can explicitly define inverse equivalences

$$\lambda g.\, (\lambda t.pr_1(g(t)),\ \lambda s.pr_2(g(s)))$$

$$\lambda h.\lambda H.\, (\lambda t.h(t),\ \lambda s.H(s))$$

and check that they compose precisely to identities. □

The extension extensionality axiom. We assume the following modified version of function extensionality: For any shape inclusion $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$ and any family $X : \{t : I \mid \psi\} \to \mathcal{U}$ such that each $X(t)$ is contractible, then, given an $a : \prod_{t:I|\phi} X(t)$, the type $\langle \prod_{t:I|\psi} X(t)|^{\phi}_{a} \rangle$ is contractible.

The usual form of function extensionality follows from this, see [RS17, Proposition 4.8] and the discussion in appendix A.2.

5.3.4 Proposition (Homotopy extension property, [RS17, Proposition 4.10]). *Let* $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$ *be a shape inclusion. Assuming extension extensionality, for any* $X : \{t : I \mid \psi\} \to \mathcal{U}$, $b : \prod_{t:I|\psi} X(t)$, $a : \prod_{t:I|\phi} X(t)$, *and* $e : \prod_{t:I|\phi} (a(t) =_{X(t)} b(t))$, *we have a map* $a' : \langle \prod_{t:I|\psi} X(t)|^{\phi}_{a} \rangle$ *and a map* $e' : \left\langle \prod_{t:I|\psi} (a'(t) = b(t))|^{\phi}_{e} \right\rangle$ *extending* e*.*

The following proposition was originally stated in the context of Segal types (🔗). We present a version for general shape inclusions (no changes to the proof are required apart from replacing the shapes (🔗)), which will enable us to prove that a function type into a fiberwise 2-Segal family is 2-Segal later on.

5.3.5 Proposition (Generalization of [RS17, Corollary 5.6]). *Let* $\{t : I \mid \phi\} \subseteq \{t : I \mid \psi\}$ *be a shape inclusion and assume that, for a type family* $C : A \to \mathcal{U}$*, where* A *is a type or shape, the induced map* $(\{t : I \mid \psi\} \to C(a)) \to (\{t : I \mid \phi\} \to C(a))$ *is an equivalence for every fiber* $C(a)$*. Then* $(\{t : I \mid \psi\} \to \prod_{x:A} C(x)) \to (\{t : I \mid \phi\} \to \prod_{x:A} C(x))$ *is an equivalence.*

5.4. Segal and Rezk types

Now that there is a way to express extensions along shape inclusions, we can repeat the mapping space construction from set-based higher category theory.

5.4.1 Definition. For a type A and $x, y : A$, we define the type of ***arrows***

$$\mathsf{hom}_A(x, y) := \left\langle \Delta^1 \to A \,|^{\partial\Delta^1}_{[x,y]} \right\rangle$$

Having a hom-type, the Segal condition is now expressed in terms of contractibility of types consisting of 2-cells witnessing candidate compositions of two composable maps. We first define the type of such 2-cells with boundaries three arrows $f : \mathsf{hom}_A(x, y)$, $g : \mathsf{hom}_A(y, z)$, $h : \mathsf{hom}_A(x, z)$ as

$$\mathsf{hom}^2_A(f, g; h) := \left\langle \Delta^2 \to A \,|^{\partial\Delta^2}_{[x,y,z,f,g,h]} \right\rangle$$

Here h is a candidate composite of f and g. Note that h is the precisely the result of evaluating at the diagonal: For $\alpha : \mathsf{hom}^2_A(f, g, h)$ and $t : \Delta^1$, we have $h(t) \equiv \alpha(t, t)$.

5.4.2 Definition. A is a ***Segal type*** if the type

$$\sum_{h:\mathsf{hom}_A(x,z)} \mathsf{hom}^2_A(f, g; h)$$

is contractible for all $x, y, z : A$, $f : \mathsf{hom}_A(x, y)$, $g : \mathsf{hom}_A(y, z)$.

Let A be a Segal type and $f : \mathsf{hom}_A(x, y)$, $g : \mathsf{hom}_A(y, z)$. By the Segal condition, we get a 2-cell $H : \Delta^2 \to A$, restricting to f and g on Λ^2_1. Then, evaluating on the diagonal $H(t, t)$ for $t : \mathbb{2}$ extracts the composite of f and g: we write $(g \circ f)(t) :\equiv H(t, t)$.

By the relevant hom^2-type being contractible, we immediately get that $g \circ f$ is propositionally unique.

5.4.3 Proposition. *If A is a Segal type, the composition operation is associative up to homotopy.*

With some extra combinatorics, we arrive at the characterization of Segal types as being local with respect to horn inclusions:

5.4.4 Theorem. *A type A is Segal if and only if it is local with respect to the horn inclusion $\Lambda^2_1 \subseteq \Delta^2$, meaning that the induced map $(\Delta^2 \to A) \to (\Lambda^2_1 \to A)$ is an equivalence.*

Isomorphisms and the Rezk condition

Isomorphisms in hom-types can be characterized in the same way that equivalences are in homotopy type theory.

5.4.5 Definition. 1. For $f : \mathsf{hom}_A(x, y)$ define the type

$$\mathsf{isiso}(f) := \Big(\sum_{g : \mathsf{hom}_A(y,x)} g \circ f = id_x \Big) \times \Big(\sum_{h : \mathsf{hom}_A(y,x)} f \circ h = id_y \Big)$$

f is an ***isomorphism*** if $\mathsf{isiso}(f)$ is inhabited.

2. For fixed $x, y : A$, denote by

$$(x \cong_A y) := \sum_{f : \mathsf{hom}_A(x,y)} \mathsf{isiso}(f)$$

the type of all isomorphisms between x and y.

Recall that this definition of isomorphisms is equivalent to f having a single two-sided inverse, but, as in homotopy type theory, we gain the following advantage:

5.4.6 Proposition. *For all $f : \mathsf{hom}_A(x, y)$, the type $\mathsf{isiso}(f)$ is a proposition.*

Proof. [RS17, Proposition 10.2] ☐

5.4.7 Construction. By path induction, it is easy to convert paths into arrows: We can define the map

$$\mathsf{idtoarr}_A : \prod_{x,y:A} \Big((x =_A y) \to (\mathsf{hom}_A(x, y)) \Big), \quad \mathsf{idtoiso}_A(a, a, \mathsf{refl}_a) :\equiv \mathsf{id}_a$$

and, since we are mapping to identities, we can specialize to isomorphisms:

$$\mathsf{idtoiso}_A : \prod_{x,y:A} \Big((x =_A y) \to (x \cong_A y) \Big), \quad \mathsf{idtoiso}_A(a, a, \mathsf{refl}_a) :\equiv (\mathsf{id}_a, ((\mathsf{id}_a, \mathsf{refl}), (\mathsf{id}_a, \mathsf{refl})))$$

5.4.8 Remark. The idea behind the map idtoarr is the same as the map idtoequiv we constructed when defining univalent universes. One can then wonder if a directed version of the univalence axiom can occur in simplicial type theory, where instead of equivalences corresponding to paths, "maps correspond to directed paths". However, the base theory in which we are working now does not include any extra axioms for universes, nor does it provide a way of defining types with specified hom-types. Recently, an extension of simplicial type theory that overcomes these problems [GWB24] was defined. We will discuss this further in (5.5.6), once we have provided the necessary background.

We can nonetheless apply the ideas from univalence and Rezk completeness to the map $\mathsf{idtoiso}_A$ internal to a type A as a definition.

5.4.9 Definition. A Segal type A is ***Rezk*** if the map $\mathsf{idtoiso}_A(x, y, -)$ is an equivalence for all $x, y : A$.

5.4.10 Remark. As this thesis is centered around a formal approach, it is worth writing down the Segal and Rezk definitions our fully formally. Then, one can see how to extract compositions, etc. using projections when formalizing simplicial type theory. For a type A, we have

$$\mathsf{isSegal}(A) :\equiv \prod_{x,y,z:A}\ \prod_{f:\mathsf{hom}_A(x,y)}\ \prod_{g:\mathsf{hom}_A(y,z)} \mathsf{isContr}\Big(\sum_{h:\mathsf{hom}_A(x,z)} \Big\langle \Delta^2 \to A \,\Big|^{\partial\Delta^2}_{[x,y,z,f,g,h]} \Big\rangle \Big)$$

$$\mathsf{isRezk}(A) :\equiv \mathsf{isSegal}(A) \times \Big(\prod_{x,y:A} \mathsf{isEquiv}(\mathsf{idtoiso}_{A,x,y}) \Big)$$

5.5. Dependent arrows and covariant families

Let $C : A \to \mathcal{U}$ be a type family over A with the corresponding projection

$$\begin{tikzcd} \sum_{x:A} C(x) \arrow[d, two heads, "pr_1"] \\ A \end{tikzcd}$$

Now let $x, y : A$ and $u : C(x)$, $v : C(y)$. Passing to hom-types gives rise to a projection map

$$\begin{tikzcd} \mathsf{hom}_{\sum_{x:A} C(x)}((x,u),(y,v)) \arrow[d, two heads, "(pr_1 \circ -) \;:\equiv\; \lambda F.\lambda t.pr_1(F(t))"] \\ \mathsf{hom}_A(x,y) \end{tikzcd}$$

To study fibrations in this directed setting, we want to work with arrows in the fibers, which should now be dependent on an arrow in the base type (For a comparison, think of the second projection $pr_2 : \prod_{z:\sum_{x:A} C(x)} C(pr_1(z))$ which depends on the first).

5.5.1 Definition. Let $C : A \to \mathcal{U}$ be a type family with $x, y : A$, $u : C(x)$, $v : C(y)$ and $f : \mathsf{hom}_A(x, y)$. The type of ***dependent arrows over*** f is defined to be

$$\mathsf{dhom}_{C(f)}(u, v) :\equiv \Big\langle \prod_{t:\mathbb{2}} C(f(t)) \Big|^{\partial\Delta^1}_{[u,v]} \Big\rangle$$

The same as above can be done for 2-cells:

For $x, y, z : A$, $u : C(x)$, $v : C(y)$, $w : C(z)$, $f : \mathsf{hom}_A(x, y)$, $g : \mathsf{hom}_A(y, z)$, $h : \mathsf{hom}_A(x, z)$, a 2-cell $\alpha : \mathsf{hom}^2(f, g; h)$ and dependent arrows $\tilde{f} : \mathsf{dhom}_{C(f)}(u, v)$, $\tilde{g} : \mathsf{dhom}_{C(f)}(v, w)$ and $\tilde{h} : \mathsf{dhom}_{C(f)}(u, w)$ we define the type

$$\mathsf{dhom}^2_{C(\alpha)}(\tilde{f}, \tilde{g}; \tilde{h}) :\equiv \Big\langle \prod_{t:\Delta^2} C(\alpha(t)) \Big|^{\partial\Delta^2}_{[\tilde{f},\tilde{g},\tilde{h}]} \Big\rangle$$

We can straightforwardly prove that, as with paths (3.4.9), arrows in the total type correspond to pairs of arrows and dependent arrows over them.

5.5.2 Notation. For $f : \mathsf{hom}_A(x,y)$ and $\tilde{f} : \mathsf{dhom}_{C(f)}(u,v)$, we denote by $(f\tilde{f})$ the map $\lambda t.(f(t), \tilde{f}(t)) : \mathsf{hom}_{\sum_{x:A} C(x)}((x,u),(y,v))$.

5.5.3 Proposition. *We have equivalences:*

1. $\mathsf{hom}_{\sum_{x:A} C(x)}((x,u),(y,v)) \simeq \sum_{f : \mathsf{hom}_A(x,y)} \mathsf{dhom}_{C(f)}(u,v)$

2. $$\sum_{H : \mathsf{hom}_{\sum_{x:A} C(x)}((x,u),(z,w))} \left(\mathsf{hom}^2_{\sum_{x:A} C(x)}((f\tilde{f}),(g\tilde{g});H)\right)$$
$$\simeq \sum_{h : \mathsf{hom}_A(x,z)} \left(\sum_{\alpha : \mathsf{hom}^2(f,g;h)} \left(\sum_{\tilde{h} : \mathsf{dhom}_{C(f)}(u,w)} \mathsf{dhom}^2_{C(\alpha)}(\tilde{f},\tilde{g};\tilde{h}) \right)\right)$$

Proof. (1.) This is just a case of "choice" (5.3.3) for the shape inclusion $\partial\Delta^1 \subseteq \Delta^1$, constant families $X :\equiv A$, $Y :\equiv (\lambda x.C(x))$ and $a :\equiv [x,y]$, $b :\equiv [u,v]$.

(2.) We can define inverse equivalences

$$\lambda(H,\mathcal{A}).\Big(\lambda t.pr_1(H(t)),\ \Big(\lambda s.pr_1(\mathcal{A}(s)),\ (\lambda r.pr_2(\mathcal{A}(r,r)),\ \lambda b.pr_2(\mathcal{A}(b)))\Big)\Big)$$
$$\lambda\Big(h,\ (\alpha,\ (\tilde{h},\tilde{\mathcal{A}}))\Big).((h\tilde{h}),\ \lambda t.(\alpha(t),\tilde{\mathcal{A}}(t)))$$

☐

5.5.4 Definition. [RS17, Definition 8.2]. A type family $C : A \to \mathcal{U}$ is ***covariant*** if for every $f : \mathsf{hom}_A(x,y)$ and $u : C(x)$, the type

$$\sum_{v : C(y)} \mathsf{dhom}_{C(f)}(u,v)$$

is contractible. Dually, C is ***contravariant*** if for every $f : \mathsf{hom}_A(x,y)$ and $v : C(y)$ the type

$$\sum_{u : C(x)} \mathsf{dhom}_{C(f)}(u,v)$$

is contractible.

5.5.5 Theorem. 🔗 *If A is Segal and $C : A \to \mathcal{U}$ is covariant, then the total type $\sum_{x:A} C(x)$ is Segal.*

5.5.6 Remark (Directed univalent universes in triangulated type theory). In a recent preprint [GWB24], Gratzer, Weinberger and Buchholtz extend simplicial type theory using modalities and embedding simplicial types in more general cubical types built out of an interval $\mathbb{I}$. With the added structure, they are able to give examples of actual Segal and Rezk types. The main construction of the paper is a "universe type of groupoids" $\mathcal{S}$, which satisfies a directed version of univalence: There is an equivalence

$$(\mathbb{I} \to \mathcal{S}) \xrightarrow{\sim} \Big(\sum_{A,B:\mathcal{S}} \mathsf{hom}_{\mathcal{S}}(A,B)\Big)$$

Composition of dependent arrows

The goal of this section is to show that, if A is Segal and $C : A \to \mathcal{U}$ is covariant, dependent arrows act like regular arrows. In particular, there is a composition operation for dependent arrows, resulting in a map over the composite of the base maps. We have formalized this result (🔗) based on an outline provided by Emily Riehl.

5.5.7 Remark. In the original proof in [RS17], this result can be extracted from the proof of (5.5.5). However, this proof is different in the formal Rzk library (which follows the "category-theoretic proof"), and thus we present an easy way to prove this independently.

We now provide an outline of the formal proof, together with the names of the statements in the Rzk file.

5.5.8 Lemma (`is-contr-comp-horn-ext-is-covariant-family-is-segal-base`)**.** *If $C : A \to \mathcal{U}$ is covariant and A is Segal, then for any $a : \Lambda^2_1 \to A$ and $c : \prod_{t:\Lambda^2_1} C(a(t))$, the type of dependent 2-cells extending c is contractible.*

5.5.9 Construction (Dependent horn, `dhorn`)**.** Let $C : A \to \mathcal{U}$ be a type family, $f : \mathsf{hom}_A(x,y)$, $g : \mathsf{hom}_A(y,z)$, $\tilde{f} : \mathsf{dhom}_{C(f)}(u,v)$, $\tilde{g} : \mathsf{dhom}_{C(f)}(v,w)$. We can define the dependent map

$$[\tilde{f},\tilde{g}]_{C([f,g])} : \prod_{t:\Lambda^2_1} C([f,g](t))$$

by $\vee$-recursion:

$$[\tilde{f},\tilde{g}]_{C([f,g])}\langle t_1,t_2\rangle :\equiv \begin{cases} \tilde{f}(t_1), & 0 \equiv t_2 \\ \tilde{g}(t_2), & t_1 \equiv 1 \end{cases}$$

5.5.10 Lemma (`dcompositions-are-dhorn-fillings`)**.** *For a type family $C : A \to \mathcal{U}$, arrows $f : \mathsf{hom}_A(x,y)$, $g : \mathsf{hom}_A(y,z)$, $h : \mathsf{hom}_A(x,z)$, a 2-cell $\alpha : \mathsf{hom}^2(f,g;h)$, and dependent arrows $\tilde{f} : \mathsf{dhom}_{C(f)}(u,v)$, $\tilde{g} : \mathsf{dhom}_{C(f)}(v,w)$, there is an equivalence*

$$\sum_{\tilde{h}:\mathsf{dhom}_{C(h)}} \mathsf{dhom}^2_{C(\alpha)}(\tilde{f},\tilde{g};\tilde{h}) \simeq \left\langle \prod_{t:\Delta^2} C(\alpha(t))\Big|^{\Lambda^2_1}_{[\tilde{f},\tilde{g}]_{C([f,g])}} \right\rangle$$

Then, the theorem `is-contr-dhom2-comp-is-covariant-family-is-segal-base` combines the two lemmas above to show that the type of dependent 2-cells extending a dependent horn is contractible. Out of this, we immediately extract dependent composition as the center of contraction (`dcomp-is-covariant-family-is-segal-base`).

6. Formalization of 2-Segal Spaces in Simplicial Type Theory

Introduction

2-Segal spaces are a generalization of Segal spaces, defined by Dyckerhoff and Kapranov [DK19]. In terms of category theory, while a (1-)Segal space is a “pre-category with compositions”, a 2-Segal space has no condition on the existence or the uniqueness of compositions, (and can be viewed as a (pre-)category with multivalued composition), but still requires higher coherence conditions to be satisfied. Since 2-Segal spaces still have associativity, they show up in many algebraic constructions, and the combination of algebraic and homotopical information is of particular use to the field of *higher algebra*. Places where 2-Segal spaces make an appearance include:

- The *Waldhausen $S_\bullet$-construction*, of particular importance to K-theory, produces a 2-Segal space.
- One can associate *Hall algebras* to 2-Segal spaces, with applications in representation theory [DK19, Chapter 8].
- An equivalence of ∞-categories between 2-Segal spaces and *invertible ∞-operads*, due to work of Walde [Wal21].

The aim of this chapter is to use the work of Dyckerhoff-Kapranov and Feller [Fel23] to introduce a meaningful definition of a *2-Segal type* in simplicial type theory. The spaces corresponding to 2-Segal types satisfy an a priori weaker condition than 2-Segal spaces, but have certain higher horn fillings themselves.

From there, we hope that we can combine this with the modal extension of simplicial type theory in [GWB24], which would allow for concrete constructions analogous to 2-Segal spaces associated to spans, or characterizations via 1-Segal conditions for path spaces, as well as applications to higher algebra in simplicial type theory, for which a possible approach is included in [GWB24].

6.1. Justification of the Semantics

Feller in [Fel23] introduces objects analogous to 2-Segal spaces in the category of simplicial *sets*, characterized by higher horn-filling conditions. The paper includes several combinatorial results relating these horns to the triangulations used by Dyckerhoff and Kapranov, and additionally exhibits higher horn inclusions as retracts of 3-horn inclusions. This can be straightforwardly generalized to $s\mathcal{S}$, meaning that 2-Segal types characterized as above really correspond to 2-Segal spaces. Analogous facts about 2-horns are used in the semantics of Segal types in [RS17, Appendix A].

6.1.1 Definition ([DK19], [Fel23])**.** For $n \geq 3$ ***2-Segal spine*** is a union of 2-simplices in Δ^n which gives a triangulation of the corresponding $(n+1)$-gon.

6.1.2 Definition ([DK19])**.** A Reedy fibrant simplicial space W is a ***2-Segal space*** if $X^{F(n)} \to X^{i_F(\mathcal{J})}$ is a trivial Reedy fibration for every 2-Segal spine inclusion $\mathcal{J} \hookrightarrow \Delta^n$.

6.1.3 Definition ([Fel23, Definition 3.1])**.** For $n \geq 3$, a subset $S \subseteq [n]$ is ***broken*** if there exist $0 \leq i < j < k < l \leq n$ such that either $i, k \in S$ and $j, l \notin S$ or vice versa.

6.1.4 Definition ([Fel23, Definition 3.3])**.** Let $n \geq 3$. A ***generalized 2-Segal horn*** is a simplicial subset $\Lambda^n_{i_1,\dots i_k}$ of Δ^n containing all faces of Δ^n except those corresponding to $i_1, \dots i_k$, with the condition that $\{i_1, \dots i_k\}$ is a broken subset of $[n]$.

6.1.5 Definition ([Fel23])**.** A simplicial space is ***quasi 2-Segal*** if $X^{F(n)} \to X^{i_F(K)}$ is a trivial Reedy fibration for every 2-Segal horn inclusion $K \hookrightarrow \Delta^n$.

6.1.6 Remark. Note that in dimension 3, the 2-Segal spaces and the 2-Segal horns coincide. So, at the very least, we know that every 2-Segal space has fillers for the two 3-dimensional 2-Segal horns.

6.1.7 Notation. Let $\mathcal{A}, \mathcal{B}$ be sets of morphisms in a category $\mathcal{C}$. We write $\mathcal{A} \pitchfork \mathcal{B}$ if $\mathcal{A}$ has the *left lifting property (LLP)* against $\mathcal{B}$, i.e., there exist lifts for all commutative squares

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "\mathcal{A} \ni f"'] & B \arrow[d, "g \in \mathcal{B}"] \\
A' \arrow[r] \arrow[ur, dotted] & B'
\end{tikzcd}$$

We also write $f \pitchfork g$ for individual morphisms f, g such that f has the LLP against g. We denote by ${}^{\pitchfork}\mathcal{A}$ (resp. $\mathcal{A}^{\pitchfork}$) the class of morphisms having the left (resp. right) lifting property against $\mathcal{A}$.

6.1.8 Definition. Let $\mathcal{C}$ be a category with small colimits and S a class of morphisms in $\mathcal{C}$. We say that S is a *(weakly)* ***saturated*** class if

1. S contains all isomorphisms.
2. S is closed under pushouts: If $(f : A \to B) \in S$ and $g : A \to C$ is any morphism in $\mathcal{C}$, then $(C \to A \coprod_B C) \in S$.
3. S is closed under transfinite composition.
4. S is closed under retracts: If $g \in S$ and f is a retract of g in the sense of (2.1.1), then $f \in S$.

The ***saturation*** $\bar{\mathcal{A}}$ of a class of morphisms $\mathcal{A}$ is the smallest saturated class containing $\mathcal{A}$.

6.1.9 Proposition (Small object argument, special case). *Assume $\mathcal{C}$ is* sSet *or* $s\mathcal{S}$*, and let $\mathcal{A}, \mathcal{B}$ be two sets of morphisms in $\mathcal{C}$. Then $\bar{\mathcal{A}} = {}^{\pitchfork}(\mathcal{A}^{\pitchfork})$.*

Additionally, we get $\mathcal{A} \pitchfork \mathcal{B} \Leftrightarrow \bar{\mathcal{A}} \pitchfork \mathcal{B}$. This is the result we use to reduce conditions to checking liftings with repsect to smaller classes of maps.

The main ingredient going into these results is the *Joyal-Tierney calculus*:

6.1.10 Definition ([JT07, Section 2]). Let X, Y be two simplicial sets. The ***box product*** $X \Box Y$ is the *simplicial space* obtained by setting

$$(X \Box Y)_{mn} := X_m \times Y_n$$

In particular, we have $i_F(X) \cong X \Box \Delta^0$.

6.1.11 Definition. Let $\mathcal{C}$ be a category with small (co)limits.

1. Let $f : A \to B, g : C \to D$ be morphisms in $\mathcal{C}$. The ***pushout product***

$$f \Box g : A \times D \coprod_{A \times C} B \times C \to B \times D$$

is the induced morphism

$$\begin{tikzcd}
A \times C \arrow[r, "f \times id_C"] \arrow[d, "id_A \times g"'] & B \times C \arrow[d] \arrow[ddr, bend left, "id_B \times g"] & \\
A \times D \arrow[r] \arrow[drr, bend right, "f \times id_D"'] & A \times D \coprod_{A \times C} B \times C \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, dotted, "f \Box g"] & \\
 & & B \times D
\end{tikzcd}$$

2. Assume that $\mathcal{C}$ is also cartesian closed. Let $g : C \to D$ and $u : X \to Y$ be morphisms in $\mathcal{C}$. We then have the ***pullback exponential***

$$\exp(g, u) : D^X \to Y^D \times_{X^C} X^D$$

induced by the pullback

$$\begin{tikzcd}
X^D \arrow[dr, dotted, "\exp(g{,}u)"] \arrow[drr, bend left, "u^D"] \arrow[ddr, bend right, "X^g"'] & & \\
 & Y^D \times_{X^C} X^D \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Y^D \arrow[d, "Y^u"] \\
 & X^C \arrow[r, "u^C"'] & Y^C
\end{tikzcd}$$

Using the adjunctions provided by the cartesian closedness of $\mathcal{C}$, we can prove that we can interchange between lifting problems:

6.1.12 Proposition ([JT07, Proposition 7.6]). *Let $\mathcal{C}$ be a cartesian closed category with small (co)limits and $f : A \to B$, $g : C \to D$, $u : X \to Y$ be morphisms in $\mathcal{C}$. Then*

$$f \Box g \pitchfork u \Leftrightarrow f \pitchfork \exp(g, u) \Leftrightarrow g \pitchfork \exp(f, u)$$

This means that we can reduce solving lifting problems to checking for certain generating morphisms, by interchanging between fibrations and cofibrations with the proposition above.

6.1.13 Proposition ([Fel23, Proposition 5.5]). *The class of 2-Segal horns in* $s\mathbf{S}\mathrm{et}$ *has the same saturated closure as the class*

$$\left\{(\Lambda^3_{0,2} \hookrightarrow \Delta^3) \Box (\partial\Delta^n \hookrightarrow \Delta^n)\right\}_{n \geq 0} \cup \left\{(\Lambda^3_{1,3} \hookrightarrow \Delta^3) \Box (\partial\Delta^m \hookrightarrow \Delta^m)\right\}_{m \geq 0}$$

This result is preserved by taking horizontal embeddings. In particular, Reedy fibrancy gives us lifts on the right side of these puhsout products, so checking that the 3-dimensional 2-Segal horn inclusions induce trivial fibrations guarantees that a simplicial space is quasi 2-Segal. By 6.1.5, every 2-Segal space is quasi 2-Segal.

6.1.14 Remark (A problem). We have not completely determined the relationship between 2-Segal horns and 2-Segal spines in simplicial spaces yet. We do know that we are formalizing at least a generalization of 2-Segal spaces that has a lot of the desired properties in common (for example, a path space criterion).

6.2. Basic definitions for 2-Segal types

We can now rewrite this equivalent definitions in simplicial type theory. The definitions and results in this section have been formalized in Rzk. One big advantage of reducing things to shape inclusions in simplicial type theory is that all the proofs of statements presented in this section are following the formal proofs for Segal types from [`05-segal-types.rzk.md`], with the original work mainly lying in the combinatorial calculations regarding the data of 3-horns and 3-simplices.

6.2.1 Definition. Let $w, x, y, z : A$, together with arrows as shown below

$$\begin{tikzcd}
& & z & \\
w \arrow[urr, "hgf"] \arrow[rrr, "gf"] \arrow[dr, "f"'] & & & y \arrow[ul, "h"'] \\
& x \arrow[uur, "hg"] \arrow[urr, "g"'] & &
\end{tikzcd}$$

and 2-cells

$$\alpha_3 : \hom^2_A(f,g;gf) \quad \alpha_2 : \hom^2_A(f,hg;hgf) \quad \alpha_1 : \hom^2_A(gf,h;hgf) \quad \alpha_0 : \hom^2_A(g,h;hg)$$

We then define the type

$$\hom^3_A(\alpha_3,\alpha_2,\alpha_1,\alpha_0) :\equiv \left\langle \Delta^3 \to A|^{\partial\Delta^3}_{[\alpha_3,\alpha_2,\alpha_1,\alpha_0]} \right\rangle$$

6.2.2 Remark. Here, the maps gf, hg and hgf are *not necessarily* the composites given by some Segal condition on the type, rather it is a notational convention added to keep track of our data (and in the context of simplicial type theory, originally appeared in the formalization of adjunctions).

6.2.3 Definition (The 3-dimensional 2-Segal horns). We define the shapes

$$\Lambda^3_{0,2} \coloneqq \left\{\langle t_1,t_2,t_3\rangle : \Delta^3 \mid t_3 \equiv 0 \vee t_1 \equiv t_2\right\}$$
$$\Lambda^3_{1,3} \coloneqq \{\langle t_1,t_2,t_3\rangle : \Delta^3 \mid t_2 \equiv t_3 \vee t_1 \equiv 1\}$$

For a visualization, functions out of 2-Segal horns correspond to the following data :

$$\begin{tikzcd}
 & & & z & \\
 & & {\alpha_1 \Leftarrow} & & \\
w \arrow[uurrr, "hgf"] \arrow[rrrr, "gf"] \arrow[dr, "f"'] & & {\alpha_3 \Uparrow} & & y \arrow[uul, "h"'] \\
 & x \arrow[urrr, "g"'] & & &
\end{tikzcd}$$

Arrows and 2-cells for a function $\Lambda^3_{0,2} \to A$

$$\begin{tikzcd}
 & & & z & \\
w \arrow[urrr, "hgf"] \arrow[dr, "f"'] & {\alpha_2 \Uparrow} & & {\alpha_0 \Leftarrow} & y \arrow[ul, "h"'] \\
 & x \arrow[uurr, "hg"] \arrow[urrr, "g"'] & & &
\end{tikzcd}$$

Arrows and 2-cells for a function $\Lambda^3_{1,3} \to A$

We can now define 2-Segal types by asking for unique fillers for both of these 2-Segal horns.

6.2.4 Definition. A type A is ***2-Segal*** if, for all $w,x,y,z : A$ and all $f : \hom_A(w,x)$, $g : \hom_A(x,y)$ and $h : \hom_A(y,z)$, both of the following conditions hold:

1. For all $gf : \hom_A(w,y)$, $hgf : \hom_A(w,z)$, $\alpha_3 : \hom^2_A(f,g;gf)$, and $\alpha_1 : \hom^2_A(gf,h;hgf)$, the type

$$\sum_{hg:\hom_A(x,z)} \Bigg(\sum_{\alpha_2:\hom^2_A(f,hg;hgf)} \Big(\sum_{\alpha_0:\hom^2_A(g,h;hg)} \hom^3_A(\alpha_3,\alpha_2,\alpha_1,\alpha_0)\Big)\Bigg)$$

   is contractible.

2. For all $hg : \hom_A(x,z)$, $hgf : \hom_A(w,z)$, $\alpha_2 : \hom^2_A(f,hg;hgf)$, and $\alpha_0 : \hom^2_A(g,h;hg)$, the type

$$\sum_{gf:\hom_A(w,y)} \Bigg(\sum_{\alpha_3:\hom^2_A(f,g;gf)} \Big(\sum_{\alpha_1:\hom^2_A(gf,h;hgf)} \hom^3_A(\alpha_3,\alpha_2,\alpha_1,\alpha_0)\Big)\Bigg)$$

   is contractible.

6.2.5 Proposition. 🔗 *A type A is 2-Segal if and only if it is local with respect to both 2-Segal horn inclusions $\Lambda^3_{0,2} \subseteq \Delta^3$, $\Lambda^3_{1,3} \subseteq \Delta^3$, i.e., the functions $(\Delta^3 \to A) \to (\Lambda^3_{0,2} \to A)$ and $(\Delta^3 \to A) \to (\Lambda^3_{1,3} \to A)$ are equivalences.*

An obvious next step would be to formally prove the fact that 2-Segal types are indeed a generalization of Segal types. For simplicial objects, a proof is given in [DK19, Proposition 2.3.4]. In our caase, the proof needs extra formalization work, showing that the 2-Segal horn inclusions are anodyne maps, analogously to [RS17, Proposition 5.20].

A. Formal Proofs Contributed to the Rzk-HoTT library

Here, we briefly present two statements, the proofs of which were contributed to the Rzk library as part of this project. We include them in a separate section, as they are standard results from homotopy type theory that are needed for the formalization of simplicial type theory later on.

A.1. Dependent pair type over a contractible base

A.1.1 Proposition. 🔗 *Let $a : A$, where A is a contractible type, and let $C : A \to \mathcal{U}$ be a type family. Then there is an equivalence*

$$\sum_{x:A} C(x) \simeq C(a)$$

This proof uses elementary facts about identity types and the transport operation (3.4.3). We first prove that there exists an equivalence in the case that a is the center of contraction in A, and then we use the fact that transport gives rise to an equivalence in the fibers, as shown below:

```
#def transport-equiv-center-fiber-total-type-is-contr-base
  ( A : U)
  ( is-contr-A : is-contr A)
  ( C : A → U)
  ( a : A)
  : Equiv
    ( Σ ( x : A) , C x)
    ( C a)
  :=
    equiv-comp
      ( Σ ( x : A) , C x)
      ( C (center-contraction A is-contr-A))
      ( C a)
      ( equiv-center-fiber-total-type-is-contr-base A is-contr-A C)
      ( equiv-transport
        ( A)
        ( C)
        ( center-contraction A is-contr-A)
        ( a)
        ( homotopy-contraction A is-contr-A a))
```

A.2. Equivalent forms of function extensionality

A.2.1 Theorem. 🔗 *Fix a universe $\mathcal{U}$. The following are equivalent:*

1. *Function extensionality: For any type family $C : A \to \mathcal{U}$ and any $f, g : \prod_{x:A} C(x)$, the function $\mathsf{htpy{-}eq}_{f,g} : (f = g) \to (f \sim g)$ defined by path induction is an equivalence.*
2. *Naive function extensionality: For any type family $C : A \to \mathcal{U}$ and any $f, g : \prod_{x:A} C(x)$, there is a function of type $(f \sim g) \to (f = g)$.*
3. *Weak function extensionality: For any type family $C : A \to \mathcal{U}$, if every fiber $A(x)$ is contractible, then the type $\prod_{x:A} C(x)$ is contractible.*

[1 ⇒ 3] was formalized by Matthias Hutzler (🔗). [1 ⇒ 2] and [2 ⇒ 3] were added after comments by Emily Riehl and Tashi Walde, noting that [2 ⇒ 3] can be immediately extracted from the proof of [1 ⇒ 3], as it only needs a map in the converse direction.

For [3 ⇒ 1], we follow the proof in [Rij22, Chapter 13], factoring through a form of the *fundamental theorem of identity types* (🔗).

Here are the three equivalent conditions from (A.2.1) written in Rzk code:

```
#def FunExt
  : U
  :=
    ( X : U)
  → ( A : X → U)
  → ( f : (x : X) → A x)
  → ( g : (x : X) → A x)
  → is-equiv (f = g) ((x : X) → f x = g x) (htpy-eq X A f g)

#def NaiveFunExt
  : U
  :=
    ( A : U) → (C : A → U)
  → ( f : (x : A) → C x)
  → ( g : (x : A) → C x)
  → ( p : (x : A) → f x = g x)
  → ( f = g)

#def WeakFunExt
  : U
  :=
    ( A : U) → (C : A → U)
  → ( is-contr-C : (a : A) → is-contr (C a))
  → ( is-contr ((a : A) → C a))
```

6.2.2 Remark. Analogous axioms are needed for extension types in simplicial type theory. As we discussed in chapter 5, Riehl and Shulman assume the strongest form, which they call *extension extensionality*, which corresponds to (A.2.1, part 2).

This axiom can be shown to imply the other possible forms in which function extensionality can be needed for extension types, but a full logical equivalence analogously to (A.2.1) has not been proven.

Bibliography

[AKS15] Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. "Univalent categories and the Rezk completion". In: *Mathematical Structures in Computer Science* 25.5 (2015), pp. 1010–1039. DOI: `10.1017/S0960129514000486`.

[AF18] David Ayala and John Francis. "Flagged higher categories". In: *Topology and quantum theory in interaction. NSF-CBMS regional conference in the mathematical sciences: topological and geometric methods in QFT, Montana State University, Bozeman, MT, USA, July 31 – August 4, 2017. Proceedings*. Providence, RI: American Mathematical Society (AMS), 2018, pp. 137–173. ISBN: 978-1-4704-4243-9; 978-1-4704-4941-4. DOI: `10.1090/conm/718/14489`.

[Bal21] Scott Balchin. A Handbook of Model Categories. Algebra and Applications. Springer Cham, 2021. ISBN: 978-3-030-75035-0. DOI: `https://doi.org/10.1007/978-3-030-75035-0`.

[Ber18] Julia E. Bergner. *The Homotopy Theory of $(\infty,1)$-Categories*. London Mathematical Society Student Texts. Cambridge University Press, 2018.

[BR12] Julia E. Bergner and Charles Rezk. *Reedy categories and the Θ-construction*. 2012. arXiv: `1110.1066v2 [math.AT]`. URL: `https://arxiv.org/abs/1110.1066`.

[BV73] J. M. Boardman and R. M. Vogt. *Homotopy invariant algebraic structures on topological spaces*. Vol. Vol. 347. Lecture Notes in Mathematics. Springer-Verlag, Berlin-New York, 1973, pp. x+257.

[BW23] Ulrik Buchholtz and Jonathan Weinberger. "Synthetic fibered $(\infty, 1)$-category theory". In: *Higher Structures* 7.1 (2023), pp. 74–165. ISSN: 2209-0606. URL: `https://higher-structures.math.cas.cz/api/files/issues/Vol7Iss1/BuchholtzWeinberger`.

[Cis19] Denis-Charles Cisinski. *Higher Categories and Homotopical Algebra*. Cambridge Studies in Advanced Mathematics.
Available at https://cisinski.app.uni-regensburg.de/CatLR.pdf. Cambridge University Press, 2019.

[Cis+] Denis-Charles Cisinski, Bastiaan Cnossen, Kim Nguyen, and Tashi Walde. "Formalization of Higher Category Theory". Notes. Work in progress, available at https://drive.google.com/file/d/1lKaq7watGGl3xvjqw9qHjm6SDPFJ2-0o/view.

[Dug08] Daniel Dugger. "A primer on homotopy colimits". Unpublished notes.
Available at https://pages.uoregon.edu/ddugger/hocolim.pdf. 2008.

[DS95] W. G. Dwyer and J. Spalinski. “Homotopy theories and model categories”. English. In: *Handbook of algebraic topology*. Amsterdam: North-Holland, 1995, pp. 73–126. ISBN: 0-444-81779-4.

[DK19] Tobias Dyckerhoff and Mikhail Kapranov. Higher Segal Spaces. Lecture Notes in Mathematics. Springer International Publishing, 2019. ISBN: 9783030271244. URL: https://doi.org/10.1007/978-3-030-27124-4.

[Fel23] Matt Feller. “Quasi-2-Segal sets”. English. In: *Tunis. J. Math.* 5.2 (2023), pp. 327–367. ISSN: 2576-7658. DOI: 10.2140/tunis.2023.5.327.

[Göd24] Jonte Gödicke. *An ∞-Category of 2-Segal Spaces*. 2024. arXiv: 2407.13357 [math.AT]. URL: https://arxiv.org/abs/2407.13357.

[GJ99] Paul G. Goerss and John F. Jardine. *Simplicial homotopy theory*. English. Vol. 174. Prog. Math. Basel: Birkhäuser, 1999. ISBN: 3-7643-6064-X.

[GWB24] Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. *Directed univalence in simplicial homotopy type theory*. 2024. arXiv: 2407.09146 [cs.LO]. URL: https://arxiv.org/abs/2407.09146.

[Hir03] Philip S. Hirschhorn. *Model categories and their localizations*. English. Vol. 99. Math. Surv. Monogr. Providence, RI: American Mathematical Society (AMS), 2003. ISBN: 0-8218-3279-4.

[HS98] Martin Hofmann and Thomas Streicher. “The groupoid interpretation of type theory”. In: *Twenty Five Years of Constructive Type Theory*. Oxford University Press, Oct. 1998. ISBN: 9780198501275. DOI: 10.1093/oso/9780198501275.003.0008. URL: https://doi.org/10.1093/oso/9780198501275.003.0008.

[Joy] Andre Joyal. “The Theory of Quasi-Categories”. Unpublished.
Available at https://math.uchicago.edu/ may/PEOPLE/JOYAL/0newqcategories.pdf.

[JT07] André Joyal and Myles Tierney. “Quasi-categories vs Segal spaces”. In: *Categories in algebra, geometry and mathematical physics*. Vol. 431. Contemp. Math. Amer. Math. Soc., Providence, RI, 2007, pp. 277–326. ISBN: 978-0-8218-3970-6; 0-8218-3970-5. DOI: 10.1090/conm/431/08278. URL: https://doi.org/10.1090/conm/431/08278.

[Joy20] Joyal’s CatLab authors. *Model categories*. https://ncatlab.org/joyalscatlab/published/Model+categories. 2020.

[KL21] Krzysztof Kapulkin and Peter Lefanu Lumsdaine. “The simplicial model of univalent foundations (after Voevodsky)”. English. In: *J. Eur. Math. Soc. (JEMS)* 23.6 (2021), pp. 2071–2126. ISSN: 1435-9855. DOI: 10.4171/JEMS/1050.

[KRW24] Nikolai Kudasov, Emily Riehl, and Jonathan Weinberger. "Formalizing the ∞-Categorical Yoneda Lemma". In: *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs*. CPP 2024. London, UK: Association for Computing Machinery, 2024, pp. 274–290. DOI: `10.1145/3636501.3636945`. URL: `https://doi.org/10.1145/3636501.3636945`.

[Lan21] Markus Land. *Introduction to infinity-categories*. English. Compact Textb. Math. Cham: Birkhäuser, 2021. ISBN: 978-3-030-61523-9; 978-3-030-61524-6. DOI: `10.1007/978-3-030-61524-6`.

[Lur09] Jacob Lurie. Higher Topos Theory (AM-170). Princeton: Princeton University Press, 2009. ISBN: 9781400830558. DOI: `doi:10.1515/9781400830558`. URL: `https://doi.org/10.1515/9781400830558`.

[Lur17] Jacob Lurie. "Higher algebra". Unpublished.
Available online at `https://www.math.ias.edu/~lurie/`. Sept. 2017.

[Lur18] Jacob Lurie. *Kerodon*. `https://kerodon.net`. 2018.

[Mac98] Saunders Mac Lane. *Categories for the working mathematician*. English. 2nd ed. Vol. 5. Grad. Texts Math. New York, NY: Springer, 1998. ISBN: 0-387-98403-8.

[MM92] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic: a first introduction to topos theory*. English. Universitext. New York etc.: Springer-Verlag, 1992. ISBN: 0-387-97710-4.

[Mar71] Per Martin-Löf. "An Intuitionistic Theory of Types". Unpublished manusctipt. 1971.

[Mar23] César Bardomiano Martínez. *Limits and exponentiable functors in simplicial homotopy type theory*. 2023. arXiv: `2202.12386 [math.CT]`. URL: `https://arxiv.org/abs/2202.12386`.

[Qui67] D. G. Quillen. *Homotopical algebra*. English. Vol. 43. Lect. Notes Math. Springer, Cham, 1967. DOI: `10.1007/BFb0097438`.

[Ras18] Nima Rasekh. *Introduction to Complete Segal Spaces*. 2018. arXiv: `1805.03131 [math.CT]`. URL: `https://arxiv.org/abs/1805.03131`.

[Ras21] Nima Rasekh. *Univalence in Higher Category Theory*. 2021. arXiv: `2103.12762 [math.CT]`. URL: `https://arxiv.org/abs/2103.12762`.

[Ree] Christopher L. Reedy. "Homotopy theory of model categories". Unpublished.
Available at https://math.mit.edu/ psh/reedy.pdf.

[Rez01] Charles Rezk. "A model for the homotopy theory of homotopy theory". English (US). In: *Transactions of the American Mathematical Society* 353.3 (2001), pp. 973–1007. ISSN: 0002-9947. DOI: `10.1090/s0002-9947-00-02653-2`.

[Rez] Charles Rezk. "Introduction to Quasicategories". Unpublished.
Available at https://rezk.web.illinois.edu/quasicats.pdf.

[Rie14] Emily Riehl. *Categorical Homotopy Theory*. New Mathematical Monographs. Cambridge University Press, 2014.

[RS17] Emily Riehl and Michael Shulman. "A type theory for synthetic ∞-categories". In: *Higher Structures* 1.1 (2017), pp. 147–224. URL: `https://higher-structures.math.cas.cz/api/files/issues/Vol1Iss1/RiehlShulman`.

[RV22] Emily Riehl and Dominic Verity. *Elements of ∞-Category Theory*. Cambridge Studies in Advanced Mathematics.
Available at https://math.jhu.edu/ eriehl/elements.pdf. Cambridge University Press, 2022.

[Rij22] Egbert Rijke. *Introduction to Homotopy Type Theory*. 2022. arXiv: `2212.11082 [math.LO]`. URL: `https://arxiv.org/abs/2212.11082`.

[Shu15] Michael Shulman. "The univalence axiom for elegant Reedy presheaves". In: *Homology Homotopy Appl.* 17.2 (2015), pp. 81–106. ISSN: 1532-0073,1532-0081. DOI: `10.4310/HHA.2015.v17.n2.a6`. URL: `https://doi.org/10.4310/HHA.2015.v17.n2.a6`.

[Shu19] Michael Shulman. *All $(\infty, 1)$-toposes have strict univalent universes*. 2019. arXiv: `1904.07004 [math.AT]`. URL: `https://arxiv.org/abs/1904.07004`.

[Sta18] The Stacks Project Authors. *Stacks Project*. `https://stacks.math.columbia.edu`. 2018.

[Ste23] Raffael Stenzel. "Univalence and completeness of Segal objects". In: *Journal of Pure and Applied Algebra* 227.4 (Apr. 2023), p. 107254. ISSN: 0022-4049. DOI: `10.1016/j.jpaa.2022.107254`. URL: `http://dx.doi.org/10.1016/j.jpaa.2022.107254`.

[Uni13] The Univalent Foundations Program. Homotopy type theory: Univalent foundations of mathematics. Institute for Advanced Study: `https://homotopytypetheory.org/book`, 2013.

[Wal21] Tashi Walde. "2–Segal spaces as invertible infinity-operads". In: *Algebraic & Geometric Topology* 21.1 (Feb. 2021), pp. 211–246. ISSN: 1472-2747. DOI: `10.2140/agt.2021.21.211`. URL: `http://dx.doi.org/10.2140/agt.2021.21.211`.

[Wei22a] Jonathan Weinberger. "A Synthetic Perspective on $(\infty,1)$-Category Theory: Fibrational and Semantic Aspects". en. PhD thesis. Darmstadt: Technische Universität Darmstadt, 2022, xxi, 177 Seiten. DOI: `https://doi.org/10.26083/tuprints-00020716`. URL: `http://tuprints.ulb.tu-darmstadt.de/20716/`.

[Wei22b] Jonathan Weinberger. *Strict stability of extension types*. 2022. arXiv: `2203.07194 [math.CT]`. URL: `https://arxiv.org/abs/2203.07194`.