Homotopy Type Theory
An Overview
(unfinished)

Nicolai Kraus

Tue Jun 19 19:29:30 UTC 2012

Preface

This composition is an introduction to a fairly new field in between of mathematics and theoretical computer science, mostly referred to as *Homotopy Type Theory*. The foundations for this subject were, in some way, laid by an article of Hofmann and Streicher [21] [1] by showing that in Intentional Type Theory, it is reasonable to consider different proofs of the same identity. Their strategy was to use groupoids for an interpretation of type theory. Pushing this idea forward, Lumsdaine [31] and van den Berg & Garner [8] noticed independently that a type bears the structure of a *weak omega groupoid*, a structure that is well-known in algebraic topology.

In recent years, Voevodsky proposed his *Univalence axiom*, basically aiming to ensure nice behaviours like the ones found in homotopy theory. Claiming that set theory has inherent disadvantages, he started to develop his *Univalent Foundations of Mathematics*, drawing a notable amount of attention from researchers in many different fields: homotopy theory, constructive logic, type theory and higher dimensional category theory, to mention the most important.

This introduction mainly consists of the first part of my first year report, which can be found on my homepage[2] (as well as updates of this composition itself). Originally, my motivation for writing up these contents has been to teach them to myself. At the same time, I had to notice that no detailed written introduction seems to exist, maybe due to the fact that the research branch is fairly new. There are several good introductions to certain aspects, but most of them require the reader to already have a good knowledge of

[1] this is not the original one, but an improved version

[2] `http://red.cs.nott.ac.uk/~ngk/`

the underlying deep-going theoretical concepts. Therefore, I hope that this introduction could be helpful for everyone interested in the subject without too much specific knowledge. I start from the very beginning and try to give a self-contained presentation. However, I believe that this attribute is somewhat inherently ill-defined. Of course, it is necessary to assume a certain level of knowledge before getting started, and here, I choose to essentially take my own, including a (however very basic) amount of type theory, homotopy theory, topology in general, and category theory.

Contents

1 Overview

In Section 2, we give very short introductions to type theory with identity types, homotopy theory, higher dimensional category theory and how they are connected. Section 3 contains a summary of the standard categorical semantics of the simply typed lambda calculus, as originally given by Lambek and Scott [26]. In Section 4, we discuss in which way the generalization described by Seely [45] can model dependently typed theories. Further, we give an introduction to the construction of homotopic models with the tools of weak factorization systems (Section 5) and model categories (Section 6), as it is done in many recent publications (including [3], [5], [6], [7], [23]). One example of a model category is the category of small groupoids, that was used by Hofmann & Streicher [21] to show that *uniqueness of identity proofs* is not implied by J (Section 7). Another model category is the one of simplicial sets, which plays a very central role in Voevodsky's model of type theory. We introduce it in Section 8. Section 9 deals with the notion of contractibility, homotopy levels, weak equivalences, univalence and concludes with a proof that univalence implies function extensionality. Finally, in Section 10, a proof of Hedberg's theorem is presented.

2 Type Theory, Homotopy Theory, Higher Category Theory and basic Intuition

We want to give a very brief introduction to the main topics we deal with.

2.1 Type Theory

Type theory, as described by per Martin-Löf ([35], [33], [34]), is an extension of the simply types Lambda calculus. Types may depend on terms of other types. Especially, if A is a type and B depends on A, then there is the *dependent sum* $\Sigma_{x:A}.B$ and the *dependent function* type $\Pi_{x:A}.B$. From the point of view of the Curry-Howard-Isomorphism, these type formers correspond to the existential and universal quantifier.

Concerning equality of terms and types, it has turned out to be very reasonable to distinguish between two main kinds: First, the *definitional equality* that type checking depends on. It is usually required to be decidable and can be used for computation. For example, β equality is often a subset of the definitional equality.

Obviously, not every interesting equality can be decidable. Instead, it is often necessary to give a concrete proof if it is claimed that two terms (or types) are equal. This is where *identity types* play a role: If a and b are of the same type A (possibly in some context), then $Id_A\,a\,b$ is a new type (in this context), the type of proofs that a equals b. The only constructor is *reflexivity*, stating that a term always equals itself. The natural elimination principle says that if a type depends on $\Sigma_{ab:A}.a \equiv b$, then we only need to construct an inhabitant for the reflexivity cases; and finally, the usual computation rule states that using the elimination rule to create a term over the reflexivity proof, we just get the same term that we have provided in the first place. The following inference rules make this precise, but there are two things to emphasize: First, we omit the assumption $\Gamma \vdash A : type$ in each rule, and second, we only state the weak form of the elimination rule. The strong one would not require the equality proof to be at the very end of the context. Of course, in the presence of dependent functions (which we always assume), the two forms are equivalent.

$$\text{Formation}\,\frac{\Gamma \vdash a, b : A}{\Gamma \vdash Id_A\,a\,b : type}$$

$$\text{Introduction}\,\frac{\Gamma \vdash a : A}{\Gamma \vdash refl_a : Id_A\,a\,a}$$

$$\text{Elimination } J\,\frac{\begin{array}{c}\Gamma \vdash P : (a, b : A) \to Id_A\,a\,b \to \textbf{Type}\\ \Gamma \vdash m : \forall a.P(a, a, refl_a)\\ \Gamma \vdash (a, b, q) : \Sigma_{a,b:A}.Id_A\,a\,b\end{array}}{\Gamma \vdash J_{P,m,(a,b,q)} : P(a, b, q)}$$

$$\text{Computation } \beta \frac{\begin{array}{c}\Gamma \vdash P : (a,b:A) \to \mathit{Id}_A\, a\, b \to \mathbf{Type} \\ \Gamma \vdash m : \forall a. P(a,a,\mathit{refl}_a) \\ \Gamma \vdash a : A\end{array}}{\Gamma \vdash J_{P,m,(a,a,\mathit{refl}_a)} =_\beta m\, a : P(a,b,q)}$$

The following is a simple, though crucial, fact:

Theorem 2.1 (equality is equ. rel.)**.** *Propositional equality is an equivalence relation, i. e. there are terms of the following types:*

1. $\mathit{refl} : \forall a\,.\,\mathit{Id}_A\, a\, a$
2. $\mathit{sym} : \forall a\, b\,.\,\mathit{Id}_A\, a\, b \to \mathit{Id}_A\, b\, a$
3. $\mathit{trans} : \forall a\, b\, c : \mathit{Id}_A\, b\, c \to \mathit{Id}_A\, a\, b \to \mathit{Id}_A\, a\, c$

Proof. The first is part of the definition, the second and the third are easily shown by applying J. $\square$

Remark 2.2. It is convenient to write p^{-1} instead of $\mathit{sym}\, p$ and $p \circ q$ instead of $\mathit{trans}\, p\, q$. Of course, the three terms are all parametrized over the type (which is A here). To improve the readability (especially in later formulas that make heavy use of reflexivity), we do not make this explicit.

The above theorem can be improved to the statement that propositional equality comes with the structure of a groupoid, or even a higher groupoid, if an appropriate formalization is given. For example, we can prove that $p \circ p^{-1}$ equals reflexivity, or that $\mathit{refl}_a \circ p$ equals p.

Sometimes, a second eliminator is used. It does not arise naturally from the definition of equality as an "inductive" type, so it has to be understood as an additional axiom:

$$\text{Streicher's Axiom } K \frac{\begin{array}{c}\Gamma \vdash a : A \\ \Gamma \vdash P : \mathit{Id}_A\, a\, a \to \mathbf{Type} \\ \Gamma \vdash m : P(\mathit{refl}_a) \\ \Gamma \vdash q : \mathit{Id}_A\, a\, a\end{array}}{\Gamma \vdash K_{a,P,m,q} : P(q)}$$

It is not difficult to check that K is actually equivalent to the principle *uniqueness of identity proofs* that states that for any valid type $\mathit{Id}_A\, a\, b$, there is at most one inhabitant:

$$\text{UIP}\,\frac{\begin{array}{c}\Gamma \vdash a, b \,:\, A \\ \Gamma \vdash p, q \,:\, \mathit{Id}_A\, a\, b\end{array}}{\Gamma \vdash \mathit{uip}_{p,q} \,:\, \mathit{Id}_{\mathit{Id}\,a\,b}\, p\, q}$$

The eliminators J and K look quite similar; and for some time, it was open whether the principle *UIP* or, equivalently, K were derivable from J. They are indeed not, as shown by Hofmann and Streicher [21]. The crucial difference is that J can eliminate if we have some type that depends on $\mathit{Id}\,a\,b$ for any a, b, while K can eliminate in the restricted case where the type depends on $\mathit{Id}\,a\,a$. The usage of K is questionable. While it was, for some time, considered a natural property, a (sufficiently strong) type theory with Voevodsky's univalence (which we will introduce later) is inconsistent, if we assume K. As McBride [36] has shown, the assumption of K allows the usage of stronger pattern matching. This might make it useful in some way. By default, the proof assistant and dependently typed programming language Agda makes use of K, while on the other hand, e. g. Coq does not.

Finally, we want to mention the *reflection rule*

$$\text{reflection}\,\frac{\begin{array}{c}\Gamma \vdash a, b \,:\, A \\ \Gamma \vdash p \,:\, \mathit{Id}_A\, a\, b\end{array}}{a = b}$$

that makes a theory *extensional* by merging definitional and propositional equality (note that, clearly, every definitional equality can be proven by reflexivity and is therefore propositionally valid). This yields (in sufficiently strong theories) undecidable typechecking and is therefore in general not at all feasible.

2.2 Homotopy Theory

A standard reference for a totally elementary and basic introduction to algebraic topology is Hatcher's book [17]. However, for the reader with a background in category theory, it might not be an economical choice. There are many other introductions available, for example [24]. Here, we only give the most basic definitions for reference, and our few paragraphs are in no way a helpful introduction to the topic.

Definition 2.3 (topological space)**.** A *topological space*, or, if no confusion is to be expected, just a *space*, is a tuple (X, τ) of a set X and a set τ of subsets of X, called a *topology* of X, that is closed under finite intersections and arbitrary unions.

Remark 2.4. In the above definition, τ contains in particular the empty set and X itself. The elements of τ are called *open* sets of X, while their complements are called *closed*. It is common to omit the topology τ in the notation and just write X instead of (X, τ).

Definition 2.5 (continuity)**.** A function $f : (X_1, \tau_1) \to (X_2, \tau_2)$ between two spaces is a map $X_1 \to X_2$ (which is, by abuse of notation, also called f) satisfying that the inverse image of each open set (i. e. element of τ_2) is again open (i. e. in τ_2).

A very special case of a topological space is a *metric space*:

Definition 2.6 (metric space)**.** A *metric space* is a tuple (X, d) of a set X and a metric (or distance function) $d : X \times X \to \mathbb{R}$, which has to have the following properties, for all x, y, z in X:

- $d(x, y) = 0$ if and only if $x = y$ (*coincidence*)
- $d(x, y) = d(y, x)$ (*symmetry*)
- $d(x, y) + d(y, z) \leq d(x, z)$ (*triangular inequality*)

Remark 2.7. A metric space (X, d) induces a topological space (X, τ), where

$$\tau = \{U \mid \forall u \in U \,.\, \exists \epsilon > 0 \,.\, \forall v \in X \,.\, d(u, v) < \epsilon \to v \in U\}.$$

Example 2.8. $\mathbb{R}$ or any subset of it is a metric space, where $d(x, y) = |x - y|$, and therefore also a topological space.

Definition 2.9 (point, path, homotopy and higher homotopy)**.** Given a topological space X. By I, we denote the unit interval $[0, 1] \subset \mathbb{R}$.

- A *point* is an element $x \in X$, or equivalently, a map $I^0 \to X$.
- A *path* is a continuous map $I \to X$. In particular, if a, b are points, then a map $p : I \to X$ with $p(0) = a, p(1) = b$ is a path from a to b.
- If a, b are points and p, q are paths from a to b, then a map $H : I^2 \to X$ with $H(0) = p$ and $H(1) = p$ and $\forall t \,.\, H(t, 0) = a \wedge H(t, 1) = b$ is a *homotopy* from p to q.

Note that we use implicit currying / uncurrying in the obvious way. Higher homotopies can be defined by the straightforward generalization as maps $I^n \to X$.

2.3 Higher Category Theory

The notion of higher categories is a generalization of ordinary category theory. Intuitively, a *2-category* is an ordinary category, enriched over the category of small categories. This means, given any two objects X, Y in a 2-category C, the "hom-set" $C(X, Y)$ is an ordinary category, the objects of which are the morphisms of C. Instead of talking about objects and morphisms, it is reasonable to call them *cells*, where the objects of C are the *0-cells*, the morphisms are the *1-cells* and the morphisms in the "hom-set" categories are the *2-cells*. A standard example of a 2-category is **Cat**, the category of all small categories, where the small categories are the 0-cells, the functors are the 1-cells and the natural transformations are the 2-cells.

In case of 2-categories, there are two possibilities: The associativity and identity laws might hold "as usual", in which case we talk about *strict 2-categories*, or only up to isomorphism, which gives us *weak 2-categories*. Similarly, a functor between 2-categories is usually weak, so that the functor laws only again hold up to isomorphism.

The situation becomes even more complicated if we look at categories of higher dimensions: *3-categories*, *4-categories*, *5-categories* and so on. An *n-category* can be described as a category enriched over (n-1)-categories. The case we are most interested in is the one of a *ω-category*, which has infinitely many levels of cells. Basically, an ω-category is an ordinary category, enriched over ω-categories. While they seem to be incredibly complicated, which is unfortunately not totally incorrect, the bright side is that there is a certain symmetry - there are ω-categories on each level, so all levels behave the same.

Fortunately, our requirements are actually a bit simpler: We are mainly interested in (ω)-*groupoids*, where every cell is an isomorphism (a 0-cell is always an isomorphism by definition).

2.4 The Connection, intuitively

The reason why homotopy theory and type theory can be brought together is that both spaces and types are instances of weak ω-groupoids.

In the case of a space, the 0-cells are the points of the space, the 1-cells the paths, the 2-cells the homotopies and in general, an n-cell is a map $I^n \to X$ with the properties listed in definition 2.9.

Van den Berg & Garner [8] and Lumsdaine [31] have independently proved that a type also forms a weak ω-groupoid: The 0-cells are terms and given two n-cells, the equality proves between them form the $n + 1$-cells.

This can be seen as an indication for the existence of a model of type theory in topological spaces; and indeed, as Voevodsky [49] has shown, such a model exists in *simplicial sets*, which is very much related to the category of *CW-complexes*, the category of "nice" topological spaces. We want to describe the most basic intuition.

A (base) type A is modelled as a space $[\![A]\!]$. Terms of this base type are just points of $[\![A]\!]$. Given two terms s, t, the type $Id_A\, a\, b$ is modelled as the space of paths between $[\![a]\!]$ and $[\![b]\!]$. Given paths p, q, the type $Id\, p\, q$ is, of course, just the space of homotopies between $[\![p]\!]$ and $[\![q]\!]$; and so on.

In this model, the eliminator J holds, while K does not, and this is exactly what we aim for. The first time I have understood this intuition completely was when I read a post by Dan Licata [27] on the HoTT blog [14], which I highly recommend. K expresses that every proof in $Id\, a\, a$ is equal to refl_a, which is not the case for paths in spaces: Obviously, not every path in some space is homotopic to the constant path. On the other hand, for J, it is enough if every element of $\Sigma_{ab:A}.Id\, a\, b$ is equal to (a, a, refl_a). In the homotopy model, this means that for any path p between $[\![a]\!]$ and $[\![b]\!]$, there is a "weak homotopy" to the constant path *const* at $[\![a]\!]$, where "weak homotopy" means a map $h : I^2 \to [\![A]\!]$ with $h(0) = p$, $h(1) = \mathit{const}$. But this is clearly true. A (canonical) choice for h is

$$h(t, s) = p((1 - t) \cdot s).$$

In my talk [25] for the FP Away Day 2012, I have tried to give some explanations of these things.

3 Semantics of the Simply Typed Lambda Calculus

In this section, we want to give an introduction to the standard categorical semantics of the simply typed lambda calculus. This content is traditional and well-known. A standard reference is [26].

To get started, consider a set S of base types, function types, finite products and coproducts. Concretely, this means that the empty type $\emptyset$ and the unit type $\{\star\}$ are types, every element of S is a type and if A, B are types, then so are $A \to B$, $A \times B$ and $A + B$. Further, every base type $o \in S$ is inhabited by a number of constants, which we denote by $\mathtt{const}_o$. Of course, for every type former, we assume the usual introduction and elimination rules.

Let $\mathbb{C}$ be a bicartesian closed category[3]. The standard categorical se-

[3]Concretely, we want $\mathbb{C}$ to have all finite products, finite coproducts and exponentials.

mantics, as, for example, described in [2], is now given as follows:

For every base type $o \in S$, let $[\![o]\!]$ be an object in $\mathbb{C}$. Let $[\![\emptyset]\!] = 0$ be the initial object and $[\![\{\star\}]\!] = 1$ the terminal object of the category. Every constant $c \in \mathtt{const}_o$ gets interpreted as a morphism $[\![c]\!] : 1 \to [\![o]\!]$. In addition, the unique inhabitant of the unite type is modelled by the identity morphism id_1.

Types. On types, the interpretation is expanded naturally :

$$\begin{array}{lcll} [\![A \times B]\!] & = & [\![A]\!] \times [\![B]\!], & \text{the categorical product} \\ [\![A + B]\!] & = & [\![A]\!] + [\![B]\!], & \text{the categorical coproduct} \\ [\![A \to B]\!] & = & [\![B]\!]^{[\![A]\!]}, & \text{the categorical exponential.} \end{array}$$

Contexts. Similarly, contexts are interpreted as

$$\begin{array}{lcll} [\![()]\!] & = & 1, & \text{where () is the empty context} \\ [\![\Gamma, x : A]\!] & = & [\![\Gamma]\!] \times [\![A]\!]. & \end{array}$$

This should not be surprising, as a context $x : A, y : B$ can be seen as the type $A \times B$ (together with a variable of that type).

Terms. As implied above, we want to interpret terms as morphisms. More precisely, the term judgement $\Gamma \vdash t : A$ will be a morphism with domain $[\![\Gamma]\!]$ and codomain $[\![A]\!]$. This is consistent to our definition of the interpretation of constants:

$$\begin{array}{lcll} [\![() \vdash c : o]\!] & = & [\![c]\!] & \\ [\![\Gamma \vdash \star : \{\star\}]\!] & = & [\![\Gamma]\!] \xrightarrow{1} 1 & \\ [\![\Gamma, x : A \vdash x : A]\!] & = & \pi_2 & \\ [\![\Gamma, x : A \vdash y : B]\!] & = & [\![\Gamma \vdash y : B]\!] \circ \pi_1 & \text{if } x \neq y \\ [\![\Gamma \vdash \mathtt{fst}\ t : A]\!] & = & \pi_1 \circ [\![\Gamma \vdash t : A \times B]\!] & \\ [\![\Gamma \vdash \mathtt{snd}\ t : B]\!] & = & \pi_2 \circ [\![\Gamma \vdash t : A \times B]\!] & \\ [\![\Gamma \vdash (r, s) : A \times B]\!] & = & \langle [\![\Gamma \vdash r : A]\!], [\![\Gamma \vdash s : B]\!] \rangle & \\ [\![\Gamma \vdash \mathtt{inl}\ r : A + B]\!] & = & \mathtt{in}_1 \circ [\![\Gamma \vdash r : A]\!] & \\ [\![\Gamma \vdash \mathtt{inr}\ s : A + B]\!] & = & \mathtt{in}_2 \circ [\![\Gamma \vdash s : B]\!] & \\ [\![\Gamma \vdash \mathtt{case}(f, g) : A + B \to C]\!] & = & [[\![\Gamma \vdash f : A \to C]\!], [\![\Gamma \vdash g : B \to C]\!]] & \end{array}$$

Here, $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ are the projections of the product, dually, $A \xrightarrow{\mathtt{in}_1} A + B \xleftarrow{\mathtt{in}_2} B$ the injections into the coproduct. Given $A \xrightarrow{f} C$ and $B \xrightarrow{g} C$, we write $[f, g]$ for the corresponding morphism $A + B \to C$. Similarly, we write $\langle h, k \rangle : C \to A \times B$, if $C \xrightarrow{h} A$ and $C \xrightarrow{k} B$ are given.

Note that the bracketing in $[\![x : A, y : B, z : C, w : D]\!] = (([\![A]\!] \times [\![B]\!]) \times [\![C]\!]) \times [\![D]\!]$ is important.

To interpret the introduction and elimination of functions, it is necessary to recall that exponentials are right adjoint to products. More precisely, for any object A, the functor $\cdot \times A : \mathbb{C} \to \mathbb{C}, B \mapsto B \times A$ is left adjoint to the functor $\cdot^A : \mathbb{C} \to \mathbb{C}, B \mapsto B^A$.

The language of category theory now offers multiple possibilities to describe what is going on. We want to be very detailed, as we believe that this in an important point:

- The adjunction $\cdot \times A \dashv \cdot^A$ comes along with its hom-set isomorphism $\Phi_{A,X,Y} : \mathbb{C}(X \times A, Y) \xrightarrow{\cong} \mathbb{C}(X, Y^A)$ (natural in all arguments). This isomorphism is sometimes (e. g. [2]) called $\texttt{curry}_{A,X,Y}$.
- *Application.* If we are given $\Gamma \vdash f : A \to B$ and $\Gamma \vdash a : A$ we can define:

$$[\![\Gamma \vdash f\, a : B]\!] = \texttt{curry}^{-1}_{[\![A]\!],[\![\Gamma]\!],[\![B]\!]}([\![\Gamma \vdash f : A \to B]\!]) \circ \langle id_{[\![\Gamma]\!]}, [\![\Gamma \vdash a : A]\!]\rangle$$

- The adjunction has a counit $\epsilon_A : \cdot^A \times A \to 1_{\mathbb{C}}$. This counit is often called $\texttt{apply}_A$. The following definition is equivalent to the one just given:

$$[\![\Gamma \vdash f\, a : B]\!] = \texttt{apply}_{[\![A]\!]}([\![B]\!]) \circ \langle [\![\Gamma \vdash f : A \to B]\!], [\![\Gamma \vdash a : A]\!]\rangle$$

  This is the definition given in [2]. Note that many authors, including [2] and Awodey [4], introduce exponentials as an object together with the morphism `apply` and its universal property. Here, we need to see the "big picture" in order to be able to discuss the generalized setting of dependent types later.
- *Abstraction.* Again, assuming we are given $\Gamma, x : A \vdash t : B$, we define

$$[\![\Gamma \vdash \lambda x.t : A \to B]\!] = \texttt{curry}_{[\![A]\!],[\![\Gamma]\!],[\![B]\!]}([\![\Gamma, x : A \vdash t : B]\!])$$

Substition/Context morphisms. Again, note that we do not make a real difference between a context such as $x : A, y : B, x : C$ and the type $A \times B \times C$ (with a variable of this type). Therefore, a *substitution* or *context morphism* is, more or less, the same as a ("generalized") term:

$$\begin{array}{lcll} [\![\Gamma \vdash \sigma : ()]\!] & = & [\![\Gamma]\!] \xrightarrow{1} 1 & \text{the same as } [\![\Gamma \vdash \star : \{\star\}]\!] \\ [\![\Gamma \vdash \sigma : \Delta, t : A]\!] & = & \langle [\![\Gamma \vdash \sigma : \Delta]\!], [\![\Gamma \vdash t : A]\!]\rangle & \end{array}$$

Lemma 3.1 (Substitution is composition)**.** *Given $\Delta \vdash t : A$ and $\Gamma \vdash \sigma : \Delta$, the equality $[\![\Delta \vdash t : A]\!] \circ [\![\Gamma \vdash \sigma : \Delta]\!] = [\![\Gamma \vdash t[\sigma] : A]\!]$ holds, where $t[\sigma]$ denotes the usual parallel substitution.*

Proof. By induction on (t, Δ). See [2], section 5.4.3. □

Theorem 3.2 (Correctness)**.** *If $\Gamma \vdash s =_{\beta\eta} t : A$, then $[\![\Gamma \vdash s : A]\!] = [\![\Gamma \vdash t : A]\!]$.*

Proof. The most interesting part of the proof is the correctness of $\beta\eta$ equality for functions. We show this part here; for the rest, see [2], section 5.4.4.

According to Lemma 3.1, we have

$$[\![\Gamma \vdash t[a/x] \,:\, B]\!] = [\![\Gamma, x : A \vdash t \,:\, B]\!] \circ \langle id_{[\![\Gamma]\!]}, [\![\Gamma \vdash a \,:\, A]\!]\rangle$$

On the other hand, we have by the definitions (for the sake of readability, we omit the indices of `curry`)

$$\begin{aligned}[\![\Gamma \vdash (\lambda x.t)\, a \,:\, B]\!] &= \texttt{curry}^{-1}\,([\![\Gamma \vdash \lambda x.t \,:\, A \to B]\!]) \circ \langle id_{[\![\Gamma]\!]}, [\![\Gamma \vdash a \,:\, A]\!]\rangle \\ &= \texttt{curry}^{-1}\,(\texttt{curry}\,([\![\Gamma, x : A \vdash t \,:\, B]\!])) \circ \langle id_{[\![\Gamma]\!]}, [\![\Gamma \vdash a \,:\, A]\!]\rangle\end{aligned}$$

This show that the β-law is given by $\texttt{curry}^{-1} \circ \texttt{curry} = id$.

Concerning η, note that

$$\begin{aligned}[\![\Gamma \vdash \lambda x.f\, x \,:\, A \to B]\!] &= \texttt{curry}\,([\![\Gamma, x : A \vdash f\, x \,:\, B]\!]) \\ &= \texttt{curry}\left(\texttt{curry}^{-1}\,([\![\Gamma \vdash f \,:\, A \to B]\!])\right)\end{aligned}$$

So, η is just the other direction of the hom-set isomorphism `curry`. □

(Co-) Inductive types. In our setting, an inductive type is the initial algebra of a functor F, while a coinductive type is the terminal coalgebra. The usual restriction is that we only deal with strictly positive functors. For example, we get the natural numbers as the initial algebra of $\mathbb{N}X = 1 + X$, lists over A by $List_A X = 1 + A \times X$ and so on. The codomain of the initial algebra (domain of the terminal coalgebra) of F is usually denoted by μF (resp. νF); the algebra itself is an isomorphism $F(\mu F) \to \mu F$ (resp. $\nu F \to F(\nu F)$ that gives us the constructors for the type while the universal property provides the eliminator, together with the β and η rule. For an explanation on that topic, see Sattler's notes [44].

Here, we do not go any further but introduce semantics of dependent types instead.

4 Semantics of Dependent Type Theory

The basic idea for moving from simple types to dependent types is surprisingly straightforward. In the above setting, we have been viewing a type as an object. However, the situation is more involved in the dependent case as a type might only be a type in a certain context, not in the empty one.

In the preceding section, a type has always been seen as a singleton context, while a term was just the same thing as a context morphism. Note that an object in $\mathbb{C}$ is just the same as an object in the slice category $\mathbb{C}/1$. The main difference in the interpretation of dependent type theory is that we can no longer use $\mathbb{C}/1$ all the time. Instead, we have to "switch between slices" whenever necessary. Assume now that $\mathbb{C}$ is not only bicartesian closed, but also locally cartesian closed and has pullbacks (or, equivalently, all finite limits).

Again, we model every context as an object of the category, in particular, the empty context is interpreted as the terminal objectand context morphisms as morphisms between them.

We interpret a type A in context Γ as a morphism with codomain $[\![\Gamma]\!]$. More precisely, the statement

$$\Gamma \vdash A \ : \ \mathit{type}$$

is modelled as a morphism

$$[\![\Gamma, x : A]\!] \xrightarrow{[\![A]\!]} [\![\Gamma]\!].$$

Here, the morphism $[\![A]\!]$ should in fact really be named $[\![\Gamma \vdash A \ : \ \mathit{type}]\!]$, we chose the simpler name for convenience. The required condition is therefore that for any type A over Γ, there has to be a morphism with codomain $[\![\Gamma]\!]$. The domain of this morphism just becomes $[\![\Gamma, x : A]\!]$.

Consequently, the statement

$$\vdash x_0 : A_0, x_1 : A_1, \ldots, x_n : A_n \ : \ \mathit{context}$$

which requires statements of the form

$$x_0 : A_0, x_1 : A_1, \ldots, x_{i-1} : A_{i-1} \vdash A_i \ : \ \mathit{type}$$

is naturally modelled as a chain of morphisms

$$[\![\Gamma_n]\!] \xrightarrow{[\![A_n]\!]} [\![\Gamma_{n-1}]\!] \xrightarrow{[\![A_{n-1}]\!]} \ldots \xrightarrow{[\![A_2]\!]} [\![\Gamma_1]\!] \xrightarrow{[\![A_1]\!]} [\![\Gamma_0]\!] \xrightarrow{[\![A_0]\!]} [\![()]\!],$$

where $\Gamma_i = x_0 : A_0, x_1 : A_1, \ldots, x_i : A_i$.

Note that a type in context Γ is nothing else but an object in the slice category over the context $[\![\Gamma]\!]$, just as it is an object in the slice over the empty context $[\![()]\!] = 1$ in the simply typed calculus. Clearly, the interpretation of a type is just the one of a "special" context morphism or substitution.

The same is true for a term. The intuition should be that a type is a projection (sometimes called a *display map*), making the context shorter by forgetting about the last entry, while a term does the opposite by constructing an inhabitant of a type, thereby prolonging the context. More precisely, the statement

$$\Gamma \vdash t \ : \ A$$

is modelled as a section of $\llbracket \Gamma, x : A \rrbracket \xrightarrow{\llbracket A \rrbracket} \llbracket \Gamma \rrbracket$, i. e. a morphism $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket \Gamma, x : A \rrbracket$ that makes

$$\begin{tikzcd}
\llbracket \Gamma, x : A \rrbracket \arrow[r, "id"] & \llbracket \Gamma, x : A \rrbracket \arrow[dl, "\llbracket A \rrbracket"] \\
\llbracket \Gamma \rrbracket \arrow[u, "\llbracket t \rrbracket"] &
\end{tikzcd}$$

commute.

Therefore, in this construction, a morphism can be both the interpretation of a type and of a term. This is not a problem and it is easy to see that if a morphism is indeed the interpretation of an inhabited type and a term, then it has to be an isomorphism.

Substitutions. Compared to the simply typed case, substitution is fairly tricky. We do not only have the substitutions for terms, but also on the type level, which we discuss first. The natural way is the solution of Seely [45], which is interpreting substitution with $\sigma : \Gamma \to \Delta$ as the pullback functor $\llbracket \sigma \rrbracket^*$. Concretely, given $\Delta \vdash A$ *: type*, we would model $\Gamma \vdash A[\sigma]$ *: type* as follows:

$$\begin{tikzcd}
\llbracket \sigma \rrbracket^* \llbracket \Delta . A \rrbracket \arrow[r, dashed] \arrow[d, dashed, "\llbracket A[\sigma] \rrbracket"] & \llbracket \Delta . A \rrbracket \arrow[d, "\llbracket A \rrbracket"] \\
\llbracket \Gamma \rrbracket \arrow[r, "\llbracket \sigma \rrbracket"'] & \llbracket \Delta \rrbracket
\end{tikzcd}$$

For a term $\Delta \vdash a$ *:* A, that is interpreted as a section of $\llbracket A \rrbracket$, we immediately get the interpretation $\llbracket \Gamma \vdash a[\sigma]$ *:* $A[\sigma] \rrbracket$ as a section of $\llbracket A[\sigma] \rrbracket$ by the pullback property.

Here are a couple of issues to be considered. First, pullbacks are clearly only unique up to isomorphism, while substitution in type theory is, for example, strictly associative. Authors such as Hofmann [19] argue that, given $\Theta \xrightarrow{\rho} \Gamma \xrightarrow{\sigma} \Delta$ and $\Gamma \vdash A$ *: type*, it is hard to ensure that $\llbracket \rho \rrbracket^* (\llbracket \sigma \rrbracket^* (\llbracket \Gamma, x : A \rrbracket))$ and $\llbracket \rho \circ \sigma \rrbracket^* (\llbracket \Gamma, x : A \rrbracket)$ are strictly equal. The situation is presented in the following diagram, where all the dotted lines are constructed as pullbacks. We shorten $\Gamma, x : A$ to $\Gamma . A$:

$$\begin{tikzcd}
\llbracket \rho \circ \sigma \rrbracket^* \llbracket \Delta . A \rrbracket \arrow[rrrd, dashed] \arrow[rdd, dashed] & & & \\
& \llbracket \rho \rrbracket^* \circ \llbracket \sigma \rrbracket^* \llbracket \Gamma . A \rrbracket \arrow[r, dotted] \arrow[d, dotted] & \llbracket \sigma \rrbracket^* \llbracket \Gamma . A \rrbracket \arrow[r, dashed] \arrow[d, dashed] & \llbracket \Delta . A \rrbracket \arrow[d, "\llbracket A \rrbracket"'] \\
& \llbracket \Theta \rrbracket \arrow[r, "\llbracket \rho \rrbracket"'] & \llbracket \Gamma \rrbracket \arrow[r, "\llbracket \sigma \rrbracket"'] & \llbracket \Delta \rrbracket
\end{tikzcd}$$

But clearly, these two upper left objects are isomorphic and it is questionable if more than that should be required in a categorical setting anyway. However, we always have to be able to recover the unique isomorphism satisfying a couple of coherence conditions.

The question how to solve these issues have given rise to a couple of suggestions. Curien [10] suggests that substitution can be modelled using an explicit pseudo-functor, while Hofmann [19] proposes *categories with attributes* (or the very similar *categories with families* [20]), which can be understood as a locally bicartesian closed category together with some additional data. The latter does the job of "choosing the correct pullbacks" and providing the coherence information.

In the case of the model in *Simplicial Sets* which is, in some sense, much more concrete, a very clean way how to solve these issues was given by Voevodsky. A good and clear presentation was given by Kapulkin, Lumsdaine and Voevodsky [22].

Dependent Sum and Function types. Again, all of the content described below was, to the best of our knowledge, first described by Seeley [45]. We first discuss dependent sums. Consider the judgements

$$\begin{aligned} \Gamma &\vdash A : \mathit{type} \\ \Gamma &\vdash a : A \\ \Gamma, x : A &\vdash B : \mathit{type} \\ \Gamma &\vdash b : A[a/x] \end{aligned}$$

and, consequently

$$\begin{aligned} \Gamma &\vdash \Sigma_{x:A}.B : \mathit{type} \\ \Gamma &\vdash (a,b) : \Sigma_{x:A}.B \end{aligned}$$

Assume we have defined the interpretation for the first four judgements. We then want to define $[\![\Gamma \vdash \Sigma_{x:A}.B : \mathit{type}]\!]$. Clearly, the codomain of this morphism has to be $[\![\Gamma]\!]$. As the domain, we choose $[\![\Gamma, x : A, y : B]\!]$ and define:

$$[\![\Gamma \vdash \Sigma_{x:A}.B : \mathit{type}]\!] = [\![\Gamma \vdash A : \mathit{type}]\!] \circ [\![\Gamma, x : A \vdash B : \mathit{type}]\!].$$

Note that in a category C that has pullbacks, there is, for any $f : X \to Y$, the pullback functor $f^* : C/Y \to C/X$ between the slices. Actually, it is dangerous to talk about "the" pullback functor, as pullbacks are only unique up to isomorphism; but as we need something like "chosen pullbacks" for the substitution anyway, we do not worry about that[4]. Then, the left

[4]actually, it is common to specify functors only up to isomorphism anyway: i.e., there is also "the" product functor.

adjoint of f^* is given by composition and usually denoted by Σ_f. The above interpretation can now be written as

$$[\![\Gamma \vdash \Sigma_{x:A}.B : \mathit{type}]\!] = \Sigma_{[\![\Gamma \vdash A : \mathit{type}]\!]}[\![\Gamma, x : A \vdash B : \mathit{type}]\!].$$

We now need to construct the interpretation of $\Gamma \vdash (a,b) : \Sigma_{x:A}.B$. From our discussion of the substitution, we know that in the diagram

$$\begin{tikzcd}
{[\![\Gamma, y : B[a/x]]\!]} \arrow[r] \arrow[d] & {[\![\Gamma, (x,y) : \Sigma_{x:A}.B]\!] = [\![\Gamma, x : A, y : B]\!]} \arrow[d, "{[\![B]\!]}"] \arrow[dl, dashed, "{\Sigma_{[\![A]\!]}[\![B]\!]}"'] \\
{[\![\Gamma]\!]} \arrow[r, "{[\![a]\!]}"] & {[\![\Gamma, x : A]\!]} \arrow[l, dashed, bend left, "{[\![A]\!]}"]
\end{tikzcd}$$

the solid square is a pullback. From the interpretation of $\Gamma \vdash b : A[a/x]$, which is a section of the leftmost morphism, we get (by composition with the uppermost morphism) the interpretation of $\Gamma \vdash (a,b) : \Sigma_{x:A}.B$.

When it comes to dependent functions, it finally becomes clear why we want $\mathbb{C}$ to be *locally* cartesian closed. An important property (for many authors actually the definition) of a locally cartesian closed category is that each pullback functor f^* has a right adjoint $f^* \dashv \Pi_f$. More precisely: Let $f : X \to Y$ be a morphism. Then, we have the following morphisms in the slice categories:

$$\Sigma_f : \mathbb{C}/Y \to \mathbb{C}/X$$
$$f^* : \mathbb{C}/X \to \mathbb{C}/Y$$
$$\Pi_f : \mathbb{C}/Y \to \mathbb{C}/X$$

with the property that

$$\Sigma_f \dashv f^* \dashv \Pi_f$$

where we have already used the first functor to interpret dependent sums. The last one will become the interpretation of dependent function types.

This connection is not surprising at all: In the simply typed setting, function types are interpreted as exponentials, the right adjoints of products. But the product with $X \to 1$ in $\mathbb{C}/1$ is just the pullback along $X \to 1$. In the dependently typed case, it could therefore have been expected that dependent types are modelled using the right adjoints of pullbacks.

We define

$$[\![\Gamma \vdash \Pi_{x:A}.B : \mathit{type}]\!] = \Pi_{[\![\Gamma \vdash A : \mathit{type}]\!]}[\![\Gamma, x : A \vdash B : \mathit{type}]\!].$$

The interpretations of the rules for terms is given by adjunction very similarly to the simply typed case. We proceed completely analogously as above:

- Given $f : A \to X$, the adjunction $f^* \dashv \Pi_f$ has an associated hom-set isomorphism. This time, we have to talk about the slice categories:

  $$\texttt{curry}_{B \xrightarrow{g} A \xrightarrow{f} X} : \mathbb{C}/A(f^* id_X, B \xrightarrow{g} A) \overset{\cong}{\longrightarrow} \mathbb{C}/X(id_X, \Pi_f B)$$

  However, one easily notices that $f^* id_X = id_A$, so that the above presentation simplifies to

  $$\texttt{curry}_{B \xrightarrow{g} A \xrightarrow{f} X} : \mathbb{C}/A(id_A, B \xrightarrow{g} A) \overset{\cong}{\longrightarrow} \mathbb{C}/X(id_X, \Pi_f B).$$

- *Application.* If we are given $\Gamma \vdash f : \Pi_{x:A}.B$ and $\Gamma \vdash a : A$, we define the application in nearly the same way as before. However, if we precede exactly as above, we get a section of $[\![B]\!] \circ [\![A]\!]$, i. e. a morphism $[\![\Gamma]\!] \to [\![\Gamma.A.B]\!]$, which is not precisely what we want. This can be fixed by by defining

  $$[\![\Gamma \vdash f\,a : B\,a]\!]$$

  to be the unique morphism $[\![\Gamma]\!] \to [\![\Gamma.B[a/x]]\!]$ that is induced by the dotted morphisms and the fact that the solid square is by definition of the substitution a pullback:

  $$\begin{tikzcd}
  {[\![\Gamma.B[a/x]]\!]} \arrow[r] \arrow[d] & {[\![\Gamma]\!]} \arrow[d, "{[\![a]\!]}"] \\
  {[\![\Gamma.A.B]\!]} \arrow[r, "{[\![B]\!]}"] & {[\![\Gamma.A]\!]} \arrow[l, dotted, bend left, "{\texttt{curry}^{-1}_{[\![\Gamma.A.B]\!] \xrightarrow{[\![B]\!]} [\![\Gamma.A]\!] \xrightarrow{[\![A]\!]} [\![\Gamma]\!]} ([\![\Gamma \vdash f : \Pi_{x:A}.B]\!])}"]
  \end{tikzcd}$$

- As before, it is possible to express this interpretation using the counit $\texttt{apply}_{[\![\Gamma \vdash A \,:\, type]\!]} : [\![A]\!]^* \left(\Pi_{[\![A]\!]} \cdot\right) \to 1_{\mathbb{C}/[\![\Gamma.A]\!]}$. Given $\Gamma \vdash f : \Pi_{x:A}.B$ and $\Gamma \vdash a : A$, these two induce the unique dotted map in the solid pullback square:

  $$\begin{tikzcd}
  {[\![A]\!]^*[\![\Gamma.\Pi_{x:A}.B]\!]} \arrow[r] \arrow[d] & {[\![\Gamma.A]\!]} \arrow[d, "{[\![A]\!]}"'] \\
  {[\![\Gamma.\Pi_{x:A}.B]\!]} \arrow[r, "{[\![\Pi_{x:A}.B]\!]}"] & {[\![\Gamma]\!]} \arrow[l, dashed, bend left, "{[\![f]\!]}"] \arrow[u, dashed, bend right, "{[\![a]\!]}"'] \arrow[ul, dotted]
  \end{tikzcd}$$

- *Abstraction.* As before, assume we are given $\Gamma, x : A \vdash t : B$. Then, we define

$$[\![\Gamma \vdash \lambda x.t : \Pi_{x:A}.B]\!] = \mathtt{curry}_{[\![A]\!],[\![\Gamma]\!],[\![B]\!]}\,([\![\Gamma, x : A \vdash t : B]\!])$$

Identity types. In this part, we show that the "naive" interpretation of type theory in locally cartesian closed categories automatically results in an *extensional* model, i. e. a model that cannot distinguish between propositional and definitional equality. The proof we give here is, to the best of my knowledge, originally due to Awodey and Warren [6]. It can also be found in Awodey's survey paper [5], Kapulkin's master thesis [23], and Arndt's and Kapulkin's and Arndt's paper [3]. I myself understand it as a justification for the (rather complicated) construction of *homotopic categorical models.*

First, note that if $\Gamma.A$ is a context, A itself is a valid type in this context. Further, there is the "diagonal map" $\delta_A : \Gamma.A \to \Gamma.A.A$, a term of this type.

The *formation rule* for identity types states that, given $\Gamma.A.A\mathit{context}$, the judgement $\Gamma, x : A, y : A \vdash \mathit{Id}_A\, x\, y : \mathit{type}$ is valid. In shortened notation, we have

$$[\![\Gamma.A.A.\mathit{Id}_A]\!] \xrightarrow{[\![\mathit{Id}_A]\!]} [\![\Gamma.A.A]\!]$$

in the model. The *introduction rule* tells us that there is, in context Γ, a term $\mathit{refl}_A : \lambda a.\mathit{Id}_A\, a\, a$. This makes sure that there is a term (or, better, context morphism)

$$[\![\Gamma.A]\!] \xrightarrow{[\![\mathit{refl}_A]\!]} [\![\Gamma.A.A.\mathit{Id}_A]\!]$$

that is a section of

$$[\![\Gamma.A.A.\mathit{Id}_A]\!] \xrightarrow{[\![\mathit{Id}_A]\!]} [\![\Gamma.A.A]\!] \xrightarrow{[\![A]\!]} [\![\Gamma.A]\!]$$

$[\![\mathit{refl}_A]\!]$ also has the (stronger) property that, composed with the display map $[\![\mathit{Id}_A]\!]$, it is equal to the term δ_A; i. e. the following commutes:

$$\begin{tikzcd}
{[\![\Gamma.A.A.\mathit{Id}_A]\!]} \arrow[d, "{[\![\mathit{Id}_A]\!]}"'] & \\
{[\![\Gamma.A.A]\!]} \arrow[r, "{[\![A^+]\!]}"] & {[\![\Gamma.A]\!]} \arrow[l, dotted, bend left, "\delta_A"] \arrow[ul, dotted, "{[\![\mathit{refl}_A]\!]}"']
\end{tikzcd}$$

Note that we shorten $\Gamma \vdash A : \mathit{type}$ to A in general, but as we really want to be able to distinguish $\Gamma \vdash A : \mathit{type}$ and $\Gamma.A \vdash A : \mathit{type}$, we write A^+ for the latter.

The *elimination rule* J says that, given a type P that depends on an identity type, i e. $\Gamma, x : A, y : A, p : \mathit{Id}_A\, x\, y \vdash P : \mathit{type}$, a term m of type $\forall a.P[a/x, a/y, \mathit{refl}_A/p]$ in context Γ, is enough to derive a term of type P in any context $\Gamma.A.A.Id_A$. This gets translated to the statement that, whenever we have a commuting square

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "\llbracket m \rrbracket"] \arrow[d, "\llbracket \mathit{refl}_A \rrbracket"'] & \llbracket \Gamma.A.A.Id_A.P \rrbracket \arrow[d, "\llbracket P \rrbracket"'] \\
\llbracket \Gamma.A.A.Id_A \rrbracket \arrow[r, "id"'] & \llbracket \Gamma.A.A.Id_A \rrbracket
\end{tikzcd}$$

the there is a diagonal filler j that makes everything commute:

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "\llbracket m \rrbracket"] \arrow[d, "\llbracket \mathit{refl}_A \rrbracket"'] & \llbracket \Gamma.A.A.Id_A.P \rrbracket \arrow[d, "\llbracket P \rrbracket"'] \\
\llbracket \Gamma.A.A.Id_A \rrbracket \arrow[r, "id"'] \arrow[ur, dotted, "j"'] & \llbracket \Gamma.A.A.Id_A \rrbracket
\end{tikzcd}$$

Note that we do not require j do be functorial or unique in any way, but for a model of type theory, it should of course be possible to get one of these diagonal fillers constructively. I is also worth noting that the lower diagram just makes sure that j is a term of the correct type, while the upper triangle represents the *computation rule* or the *β-rule* of the identity type: If we use J with some term m to construct an inhabitant of $P[a/x, a/y, \mathit{refl}_a/p]$, then this is just m itself (in type theory, definitionally, in the model, equal as hom-set morphisms).

But here is a problem: We do not want this diagonal filler j to exist in general, we only need it if the right vertical morphism $\llbracket P \rrbracket$ is a type. If we require the existence of j just for any morphism on the right side as long as we have $\llbracket \mathit{refl}_A \rrbracket$ in the left side, it would in particular exist in the following diagram:

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "id"] \arrow[d, "\llbracket \mathit{refl}_A \rrbracket"'] & \llbracket \Gamma.A \rrbracket \arrow[d, "\llbracket \mathit{refl}_A \rrbracket"'] \\
\llbracket \Gamma.A.A.Id_A \rrbracket \arrow[r, "id"'] & \llbracket \Gamma.A.A.Id_A \rrbracket
\end{tikzcd}$$

Now, commutativity of the diagram implies that $\llbracket \mathit{refl}_A \rrbracket$ is an isomorphism. The sources cited above immediately conclude that the model is extensional. However, it took me quite some time to really understand it; after all, $\mathit{refl}_A : A \to \Sigma_{x:A,y:A}.Id_A\, x\, y$ *is* a "type theoretical" isomorphism. "Type theoretical isomorphism" (or "bijection" [11]) means that there is a term t and terms of type $\forall a.a \equiv t \circ \mathit{refl}_A a$ as well as $\forall p.p \equiv \mathit{refl}_A \circ tp$.[5] However, these "type theoretic" isomorphisms are not necessarily modelled by

[5]This is equivalent to saying that refl_A is a weak equivalence and, assuming univalence, A and $\Sigma_{x:A,y:A}.Id_A\, x\, y$ are actually equal types.

isomorphisms! After all, their compositions are only in a very weak sense equal to the identity.

Assume we have $\Gamma \vdash s_0, s_1 : A$ and $\Gamma \vdash p' : Id\, s_0\, s_1$. We set $p = (s_0, s_1, p')$. Let us draw another diagram:

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "\llbracket \mathit{refl}_A \rrbracket"] \arrow[d, "\delta"'] & \llbracket \Gamma.A.A.Id_A \rrbracket \arrow[dl, "\llbracket Id_A \rrbracket"'] \\
\llbracket \Gamma.A.A \rrbracket & \llbracket \Gamma \rrbracket \arrow[l, "\llbracket (s_0, s_1) \rrbracket"] \arrow[u, "\llbracket p \rrbracket"']
\end{tikzcd}$$

Note that the labelling of the bottom vertical arrow is slightly inaccurate, better (but less readable) would be $\llbracket A \rrbracket^*(\llbracket s_1 \rrbracket) \circ \llbracket s_0 \rrbracket$.[6] The commutativity of the lower diagram means that $p = (s_0, s_1, p')$ is really a proof that s_0 and s_1 are equal terms. Commutativity of the upper triangle is just the property of refl_A discussed above. But now, we have (if we omit the substitution again which does not change anything) $\llbracket (s_0, s_1) \rrbracket = \delta \circ \llbracket \mathit{refl}_A \rrbracket^{-1} \circ \llbracket p \rrbracket$ and therefore $\llbracket s_0 \rrbracket = \llbracket \mathit{refl}_A \rrbracket^{-1} \circ \llbracket p \rrbracket = \llbracket s_1 \rrbracket$. Summarized: If there is a proof that two terms are propositionally equal, then they are modelled by the same morphism.

It is therefore not possible to model proper intensional type theory in this naiv way. The problem is that we have required the diagonal filler j to exist whenever the vertical left arrow is a reflexivity proof; a straightforward idea to fix it is restricting this requirement to a class of morphisms containing every interpretation of a type. However, it turns out to be very hard to give a good nontrivial characterization of such a class; and actually, all the work on *homotopic-theoretical models* can to some extend be seen as attempts to find those characterizations.

5 Homotopic Models

The main sources for this section are van den Berg & Garner [7], Arndt & Kapulkin [3] and Awodey & Warren [6]. For some concepts, the $n$lab [48] is very useful.

5.1 Weak Factorization Systems

Weak factorization systems provide a useful setting for models of identity types. As described in the section about semantics, the critical point is to find a suitable subclass of morphisms that have the properties of types, and weak factorization system address exactly at this issue. It also works

[6]$\Gamma.A.A$ is isomorphic to $\Gamma.A \times A$. This justifies our simplification.

the other way round: As shown by Gambino & Garner [16], the classifying category of a type theory with identity types admits always a (nontrivial) weak factorization system.

We use the definitions stated in [39]. The concepts are well-known and wildly accepted, but to the best of my own knowledge, they are originally due to Bousfield [9].

It will turn out later that a basic requirement for identity types is the following *lifting property.*

Definition 5.1 (Lifting Property)**.** Let c, f be morphisms in a category. c has the *left lifting property* with respect to f (equivalently, f has the *right lifting property* with respect to c) if, for any commutative square

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "c"'] & X \arrow[d, "f"] \\
B \arrow[r] & Y
\end{tikzcd}$$

a *diagonal filler*, i.e. a morphism $j : B \to X$, exists, making the whole diagram commutative. This filler does not have to be unique (though this would be a useful property later).

Having this concept in hand, we are able to define the mentioned systems:

Definition 5.2 (weak factorization system)**.** Given a category C, a *weak factorization system* on C is a pair $(\mathfrak{L}, \mathfrak{R})$ of sets of morphisms of C such that

(W_1) every morphism in C can be written as $p \circ i$, where $p \in \mathfrak{R}, i \in \mathfrak{L}$ (not necessarily unique)

(W_2) every morphism in $\mathfrak{L}$ has the left lifting property with respect to every morphism in $\mathfrak{R}$

(W_3) $\mathfrak{L}$ and $\mathfrak{R}$ are maximal with this property (i.e. we cannot add any morphisms without violating the above requirements)

Example 5.3. A basic (but nonconstructive) example of a weak factorization system on the category of sets is (*monos*, *epis*), i.e. the set of injective (monomorphisms) and surjective maps (epimorphisms). The factorization of $f : A \to B$ is given by

$$\begin{tikzcd}
A \arrow[rr, "f"] \arrow[dr, dashed, hook, "a \mapsto (in_1 a, f(a))"'] & & B \\
& (1+A) \times B \arrow[ur, dashed, two heads, "(a,b) \mapsto b"'] &
\end{tikzcd}$$

Note that the somewhat nasty 1+ is necessary as there would be a problem if $A = \emptyset \neq B$ otherwise. For the lifting j, given an injective c and a surjective f in

$$\begin{tikzcd} A \arrow[r, "u"] \arrow[d, "c"'] & X \arrow[d, "f"] \\ B \arrow[r, "v"'] & Y \end{tikzcd}$$

define $j(b) = u(a)$ if $a = c(a)$ and, if such an a does not exist, $j(b) = x$ for some x that satisfies $f(x) = v(b)$. Note that this is possible if and only if the Axiom of Choice holds true. Finally, the maximality condition is clearly satisfied.

This can also be done in a constructive way by requiring that all the monos and epis are split (and carry information about the corresponding retraction resp. section), thus essentially making a (deterministic) construction of the diagonal filler possible.

As an easy exercise and because it is important for further explanations, we prove the following:

Lemma 5.4. *Given a weak factorization system* $(\mathfrak{L}, \mathfrak{R})$*, the class* $\mathfrak{R}$ *is closed under pullbacks (whenever they exist).*

Proof. Let $f : B \to A \in \mathfrak{R}$ and $\sigma : X \to A$ any morphism. Then, for $Y = X \times_A B$, we have to show that in the pullback square

$$\begin{tikzcd} Y \arrow[r, "\tau"] \arrow[d, "g"'] & B \arrow[d, "f"] \\ X \arrow[r, "\sigma"'] & A \end{tikzcd}$$

the morphism g is in $\mathfrak{R}$. Therefore, let $c : S \to T \in \mathfrak{L}$ be any morphism and $s : S \to Y$, $t : T \to X$ be morphisms that make the left square and therefore the whole diagram commute:

$$\begin{tikzcd} S \arrow[r, "s"] \arrow[d, "c"'] & Y \arrow[r, "\tau"] \arrow[d, "g"'] & B \arrow[d, "f"] \\ T \arrow[r, "t"'] & X \arrow[r, "\sigma"'] & A \end{tikzcd}$$

As $f \in \mathfrak{R}$, there exists a diagonal filler:

$$\begin{tikzcd} S \arrow[r, "s"] \arrow[d, "c"'] & Y \arrow[r, "\tau"] & B \arrow[d, "f"] \\ T \arrow[r, "t"'] \arrow[urr, dashed, "i"] & X \arrow[r, "\sigma"'] & A \end{tikzcd}$$

Together with the pair (i, t), the property of the pullback guarantees that there is a morphism j:

$$\begin{tikzcd}
S \arrow[r, "s"] \arrow[d, "c"'] & Y \arrow[r, "\tau"] \arrow[d, "g"] & B \arrow[d, "f"] \\
T \arrow[r, "t"'] \arrow[ur, dotted, "j"] & X \arrow[r, "\sigma"'] & A
\end{tikzcd}$$

By our construction, we have $g \circ j = t$, so the only thing to check is whether s equals $j \circ c$. We know that $g \circ s = g \circ j \circ c$ and that $\tau \circ s = i \circ c = \tau \circ j \circ c$, so the pairs $(\tau \circ s, g \circ s)$ and $(\tau \circ j \circ c, g \circ j \circ c)$ are equal. Because of the pullback property, each of the two pairs provides a unique morphism $S \to Y$ making everything commutative, but for the first pair, this is obviously s and for the second pair, this is $j \circ c$. Consequently, they are equal and we are done. $\square$

The above lemma enables us to interpret substitution of types as pullbacks, as described in the section about semantics of dependent type theory.

5.2 Identity types in Weak Factorization Systems

I have learnt the content of this subsection by reading the (highly recommended) survey article [5]. Unfortunately, it is not very detailed, so I try to give a slightly longer explanation.

From now on, let us write A^I as short-hand for $\Sigma_{a,b:A}.Id_A\, a\, b$. Clearly, if A is a type in context Γ, then so is A^I[7].

So, let us now discuss the interpretation of identity types. Given a bicartesian closed categoriy $\mathbb{C}$ with a weak factorization system $(\mathfrak{L}, \mathfrak{R})$, assume we have interpreted everything apart from identity types as described in the semantics section. If $\Gamma \vdash A$ *: type*, then there is the context morphism $\delta_A : \Gamma.A \to \Gamma.A.A$. In $\mathbb{C}$, the morphism $[\![\delta]\!]_A$ can, according to the laws of a weak factorization system, be written as

$$[\![\delta_A]\!] = [\![\Gamma.A]\!] \xrightarrow{c \in \mathfrak{L}} X \xrightarrow{f \in \mathfrak{R}} [\![\Gamma.A.A]\!].$$

Assume we have a possibility to choose one of the (possible multiple) factorizations in a coherent way. Then, we can choose to model $\Gamma.A^I$ by X, and c will be the interpretation of $\lambda a\,.\,(a, a, \mathit{refl}_a)$, while f is the interpretation of the equality type (that depends on $\Gamma.A.A$).

For an easier notation, let us write r_A instead of $\lambda a : A\,.\,(a, a, \mathit{refl}_a)$.

[7]This is usually the notation for the *path object* in abstract homotopy theory of model categories; this coincidence is, of course, not random!

The property of the weak factorization system makes sure that we can interpret the eliminator. Consider a type P that depends on A^I in context Γ and assume we have a term m of type $\forall a.P(a, a, \mathit{refl}_a)$ as in the commutative diagram

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "\llbracket m \rrbracket"] \arrow[d, "\llbracket r_A \rrbracket"'] & \llbracket \Gamma.A^I.P \rrbracket \arrow[d, "\llbracket P \rrbracket"] \\
\llbracket \Gamma.A^I \rrbracket \arrow[r, "1"] & \llbracket \Gamma.A^I \rrbracket
\end{tikzcd}$$

This diagram represents exactly the assumptions of the J rule. We know that the left and the right morphism are in $\mathfrak{L}$ resp. in $\mathfrak{R}$. Therefore, the properties of the weak factorization system guarantee the existence of a morphism j making the diagram commutative:

$$\begin{tikzcd}
\llbracket \Gamma.A \rrbracket \arrow[r, "\llbracket m \rrbracket"] \arrow[d, "\llbracket r_A \rrbracket"'] & \llbracket \Gamma.A^I.P \rrbracket \arrow[d, "\llbracket P \rrbracket"] \\
\llbracket \Gamma.A^I \rrbracket \arrow[r, "1"] \arrow[ur, "j"'] & \llbracket \Gamma.A^I \rrbracket
\end{tikzcd}$$

If we have a coherent possibility to choose the filler j, we can use it as the interpretation of the elimination rule. Note that the upper triangle represents the computation rule, stating $j \circ \llbracket r_A \rrbracket = \llbracket m \rrbracket$.

5.3 Homotopy Theoretic Models of Identity Types

Let us state the above discussion in form of a theorem. Is is due to Awodey & Warren [6] and makes use of some abstract homotopy theory (in particular, *path objects*, which we do not repeat here; a good source is [12]):

Theorem 5.5. *Let $\mathbb{C}$ be a finitely complete category with a weak factorization system and a functorial choice $(\cdot)^I$ of path objects in C, and all of its slices, which is stable under substitution. I.e., given any $A \to \Gamma$ and $\sigma : \Gamma' \to \Gamma$,*

$$\sigma^*(A^I) \cong (\sigma^* A)^I.$$

Then $\mathbb{C}$ is a model of a form of Martin-Löf type theory with identity types.

Note that A^I is now defined in both the cases that A is a type and that A is an object in a category, which will hopefully not lead to confusion. The intuition is that the former should be modelled by the latter.

Here is a diagram that illustrates the theorem. Given an $f \in \mathfrak{R}$ and a morphism σ, which are painted as solid arrows, we can construct the pullbacks (dashed arrows) and factorizations of the diagonals (dotted arrows):

$$\begin{tikzcd}
\sigma^*A \times_{\Gamma'} \sigma^*A & & & & \\
 & & (\sigma^*A)^I \arrow[ull, dotted] & & \\
 & & & \sigma^*A \arrow[ul, dotted, "r_{\sigma * A}"'] \arrow[r, dashed] \arrow[dd, dashed] & \Gamma' \arrow[dd, "\sigma"] \\
 & \sigma^*(A^I) \arrow[uur, squiggly, no head] \arrow[urrr, dashed] \arrow[dddr, dashed] & & & \\
 & & & A \arrow[r, "f"] & \Gamma \\
 & & & A \times_\Gamma A \arrow[u, dashed] \arrow[r, dashed] & A \arrow[u, "f"'] \\
 & & A^I \arrow[ur, dotted] & & \\
 & A \arrow[ur, dotted, "r_A"] & & &
\end{tikzcd}$$

As shown in the diagram, $\sigma^*(A^I)$ and $(\sigma^*A)^I$ have to be isomorphic in order to fulfil the assumptions of the theorem.

The proof given in [6] is basically a summary of our explanations in the previous sections.

Remark 5.6. At first sight, one might wonder if the choices of j in Awodey & Warren's theorem have to fulfil some coherence conditions. The authors do not mention any, but personally, I am still not completely sure about this.

6 Model Categories

In this section, we want to provide some background on *model categories*, a structure that can be found in many mathematical constructions and has two weak factorization systems built-in. They were first mentioned by Quillen [42]. Many authors ([3], [5], [6], [7], [23], [30]) make use of model categories, but in fact, one of the two weak factorization systems is never used. However, if something has the structure of a weak factorization system, it nearly always is a model category as well, so not much harm is done if the stronger notion is assumed. The advantage is that model categories are a well-established concept in mathematics.

We first state the definition given in the *n*lab [38]:

Definition 6.1 (Model Category)**.** A *model structure* on a category C consists of three distinguished classes of morphisms of C, namely the cofibrations $\mathfrak{C}$, the fibrations $\mathfrak{F}$ and the weak equivalences $\mathfrak{W}$, satisfying:

(M_1) *2-out-of-3*: If, for any two composable morphisms f, g, two of the three morphisms $f, g, g \circ f$ are weak equivalences, then so is the third.

(M_2) there are two weak factorization systems, $(\mathfrak{C}, \mathfrak{W} \cap \mathfrak{F})$ and $(\mathfrak{C} \cap \mathfrak{W}, \mathfrak{F})$.

A *model category* is a complete (all small limits exist) and cocomplete (all small colimits exist) category that carries a model structure.

Like most authors, we call a map that is a fibration and a weak equivalence at the same time an *acyclic fibration* (not that *trivial fibration* is also common), similarly, a cofibration that is a weak equivalence is called an *acyclic cofibration.* If $A \to 1$ is a fibration, then the object A is called *fibrant.* If $0 \to B$ is a cofibration, B is called *cofibrant.*

We allow us to make three comments here, all of which are well-known and are, for example, stated in [32].

Example 6.2. For any category C, let two of the three classes $\mathfrak{F}, \mathfrak{C}, \mathfrak{W}$ be the class of all morphisms and let the third be the class of all isomorphisms. This gives us three different model structures on C. If C has all small limits and colimits, these construction forms a model category.

Example 6.3. The product of model categories is, in the obvious way, a model category.

Remark 6.4. The concept of a model category is self-dual.

Nontrivial examples include the categories of groupoids, simplicial sets and topological spaces, which will be discussed later.

As we will need it soon, we want to state the the following proposition ([12] Proposition 3.14):

Proposition 6.5. *Let C be a model category. Then the following statements are true:*

(i) *$\mathfrak{C}$ is closed under* cobase change*, i.e. if c is a cofibration in the following pushout diagram, then so is c':*

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "c"'] & C \arrow[d, dashed, "c'"] \\
B \arrow[r, dashed] & D
\end{tikzcd}$$

(ii) *$\mathfrak{W} \cap \mathfrak{C}$ is closed under* cobase change*, i.e. if c is an acyclic cofibration in the diagram above, then so is c'.*

(iii) *$\mathfrak{F}$ is closed under* base change*, which is the dual statement of (i), i.e. the pushout is replaced by a pullback.*

(iv) *$f \cap \mathfrak{W}$ is closed under* base change*.*

Proof. The statement (ii) can be proved analogously to (i) and $(iii), (iv)$ are clearly dual to $(i), (ii)$, so we only prove (i). To prove that c' in the given diagram is a cofibration, let f be an acyclic fibration.

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "c"'] & C \arrow[r, "u"] \arrow[d, dashed, "c'"] & E \arrow[d, "f"] \\
B \arrow[r, dashed, "s"'] & D \arrow[r, "t"'] & F
\end{tikzcd}$$

As c is a cofibration and f an acyclic fibration, there exists a lifting $j : B \to E$. Together with u and the pushout property, j implies that there is a morphism $j' : D \to E$. While [12] concludes the proof by stating that j' is the required lifting, the authors do not find it obvious that both triangles commute.

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "c"'] & C \arrow[r, "u"] \arrow[d, dashed] & E \arrow[d, "f"] \\
B \arrow[r, dashed, "s"'] \arrow[urr, dotted, "j"] & D \arrow[r, "t"'] \arrow[ur, dotted, "j'"'] & F
\end{tikzcd}$$

While the commutativity of the upper triangle, i.e. $u = j' \circ c'$ is indeed a direct consequence of the pullback property, it is less clear that $t = f \circ j'$. However, note that $f \circ j' \circ c' = f \circ u = t \circ c'$ and $f \circ j' \circ s = f \circ j = t \circ s$ do directly follow from the commutativity of the diagram, where commutativity of the lower triangle is not assumed. We have therefore shown that the pairs $(C \xrightarrow{f \circ j' \circ c'} F, B \xrightarrow{f \circ j' \circ s} F)$ and $(C \xrightarrow{t \circ c'} F, B \xrightarrow{t \circ s} F)$ are identical, which means that they induce (by the pushout property) the same unique morphism $D \to F$, but as those morphisms are $f \circ j'$ respective t, those are equal. $\square$

We want to state another often used definition, used, for example, in [12] and [32] (however, both times slightly differently). One thing should be mentioned before:

Definition 6.6 (Retract)**.** If X, Y are two objects in a category C, we say that Y is a *retract* of X if there exists maps $Y \xrightarrow{s} X \xrightarrow{r} Y$ with $r \circ s = id_Y$. If f, g are maps, we call g a *retract* of f if this holds true in the arrow category $C^{\rightarrow}$, i.e. if there is a commuting diagram

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "g"'] & W \arrow[r, "p_1"] \arrow[d, "f"'] & A \arrow[d, "g"'] \\
B \arrow[r, "i_2"'] & Z \arrow[r, "p_2"'] & B
\end{tikzcd}$$

satisfying $p_1 \circ i_1 = id_A$ and $p_2 \circ i_2 = id_B$.

Definition 6.7 (Model Category, alternative definition)**.** As above, a *model structure* on C consists of three classes $\mathfrak{C}, \mathfrak{F}, \mathfrak{W}$, so that:

(N_1) $\mathfrak{C}, \mathfrak{F}, \mathfrak{W}$ are all closed under composition and include all identity maps

(N_2) *2-out-of-three* is satisfied as above: If two of f, g, $g \circ f$ are in $\mathfrak{W}$, then so is the third

(N_3) $\mathfrak{C}, \mathfrak{F}, \mathfrak{W}$ are also closed under retracts, i.e. if g is a retract of f and f is in one of the classes, then g is in the same class

(N_4) if $c \in \mathfrak{C}$ and $f \in \mathfrak{F}$, then c has the left lifting property with respect to f if at least one of them is also in $\mathfrak{W}$

(N_5) any morphism m can be factored in two ways:

$$A \xrightarrow{m} B = A \xrightarrow{c \in \mathfrak{C} \cap \mathfrak{W}} X \xrightarrow{f \in \mathfrak{F}} B = A \xrightarrow{c' \in \mathfrak{C}} X' \xrightarrow{f' \in \mathfrak{F} \cap \mathfrak{W}} B$$

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

Remark 6.8. There are a couple of variants to these definitions. For example, Dwyer & Spalinski [12] only requires the existence of finite limits and colimits. Also, Hovey [32] wants the factorizations to be functorial. The *n*lab [38] states: "Quillens original definition required only finite limits and colimits, which are enough for the basic constructions. Colimits of larger cardinality are sometimes required for the small object argument, however."

As an exercise, we want to prove that each of the above definitions can be replaced by the other.

Proposition 6.9. *The definitions 6.1 and 6.7 are equivalent.*

Proof. For the first direction, we have to show that (M_1) and (M_2) imply $(N_1) - (N_5)$. For (N_2), (N_4) and (N_5), there is nothing to do.

Concerning (N_1), the maximality condition in the definition of a weak factorization system clearly implies that each class contains all identity maps. The closedness under composition of $\mathfrak{W}$ follows from *2-out-of-3*. For the rest, let f_1, f_2 be composable fibrations. To show that $f_2 \circ f_1$ is a fibration as well, assume that c is any acyclic cofibration. Given a commuting diagram

$$\begin{tikzcd}
A \arrow[r, "s"] \arrow[dd, "c"'] & X \arrow[d, "f_1"] \\
 & Z \arrow[d, "f_2"] \\
B \arrow[r, "t"'] & Y
\end{tikzcd}$$

In this situation, (M_2), applied on c and f_2, guarantees the existence of a lift $B \to Z$. Using this lift, we can apply the same argument again on c and f_1, obtaining a lift $B \to X$ as required. Using the maximality condition of weak factorization systems again, we conclude that $f_2 \circ f_1$ is indeed a fibration. The rest of (N_1) follows analogously.

Concerning (N_3), we first show that $\mathfrak{F}$ is closed under retracts. Assume $f \in \mathfrak{F}$ and g is a retract of f. Let c be any acyclic cofibration. We have to show that the diagonal filler exists in

$$\begin{tikzcd}
A \arrow[r, "s"] \arrow[d, "g"'] & X \arrow[d, "c"] \\
B \arrow[r, "t"'] & Y
\end{tikzcd}$$

The fact that g is a retract of f yields an extension of this diagram, namely

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "g"'] & W \arrow[r, "p_1"] \arrow[d, "f"'] & A \arrow[r, "s"] \arrow[d, "g"'] & X \arrow[d, "c"] \\
B \arrow[r, "i_2"'] & Z \arrow[r, "p_2"'] & B \arrow[r, "t"'] & Y
\end{tikzcd}$$

where $p_1 \circ i_1 = id_A$ and $p_2 \circ i_2 = id_B$. As f is a fibration, we get a filler $j : Z \to A$ and consequently, $j \circ i_2$ is a valid filler for the original square. Using the maximality property of $\mathfrak{F}$ proves $g \in \mathfrak{F}$. If we require c only to be a cofibration, not necessarily acyclic, the same argumentation shows that

$\mathfrak{F} \cap \mathfrak{W}$ is closed under retracts. Dually, $\mathfrak{C}$ and $\mathfrak{W} \cap \mathfrak{C}$ are both closed under retracts.

Surprisingly, closeness of $\mathfrak{W}$ under retracts is a bit tricky. This case is actually the reason why we stated proposition 6.5 above. Assume $f \in \mathfrak{W}$ and g is a retract of f. Using (M_2), we factorize g as $g = p \circ i$, where $i \in \mathfrak{C}$ and $p \in \mathfrak{W} \cap \mathfrak{F}$, yielding the following commutative diagram:

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "i"'] & W \arrow[r, "p_1"] \arrow[dd, "f"] & A \arrow[d, "i\in\mathfrak{C}"] \\
K \arrow[d, "p"'] & & K \arrow[d, "p\in\mathfrak{W}\cap\mathfrak{F}"] \\
B \arrow[r, "i_2"'] & Z \arrow[r, "p_2"'] & N
\end{tikzcd}$$

We now take the pullback of p along p_2. As shown in proposition 6.5, the map $Z \times_B K \to Z$ is in $\mathfrak{W} \cap \mathfrak{F}$:

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "i"'] & W \arrow[r, "p_1"] & A \arrow[d, "i\in\mathfrak{C}"] \\
K \arrow[d, "p"'] & Z \times_B K \arrow[r, dashed, "v"] \arrow[d, dashed, "p'\in\mathfrak{W}\cap\mathfrak{F}"] & K \arrow[d, "p\in\mathfrak{W}\cap\mathfrak{F}"] \\
B \arrow[r, "i_2"'] & Z \arrow[r, "p_2"'] & N
\end{tikzcd}$$

Because of the pullback property, the pairs $(id_K, i_2 \circ p)$ and $(f, i \circ p_1)$ induce the following maps

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "i"'] & W \arrow[r, "p_1"] \arrow[d, dotted, "i'\in\mathfrak{W}"] & A \arrow[d, "i\in\mathfrak{C}"] \\
K \arrow[d, "p"'] \arrow[r, dotted, "u"] & Z \times_B K \arrow[r, dashed, "v"] \arrow[d, dashed, "p'\in\mathfrak{W}\cap\mathfrak{F}"] & K \arrow[d, "p\in\mathfrak{W}\cap\mathfrak{F}"] \\
B \arrow[r, "i_2"'] & Z \arrow[r, "p_2"'] & N
\end{tikzcd}$$

where $p' \circ i' = f$ and $v \circ u = id_K$. The reason for i' being in $\mathfrak{W}$ is *two-out-of-three*, as p' and f are both weak equivalences. We now focus on the top half of the diagram. We can factorise i' as $i' = W \xrightarrow{i''\in\mathfrak{C}\cap\mathfrak{W}} X \xrightarrow{p''\in\mathfrak{W}\cap\mathfrak{F}} Z \times_B K$, where *two-out-of-three* is again the reason why we may assume that both i'' and p'' are weak equivalences. Further, by the lifting property, we get the morphism j:

$$\begin{tikzcd}
A \arrow[rr, "i_1"] \arrow[dd, "i\in\mathfrak{C}"'] & & W \arrow[r, "p_1"] \arrow[dl, dotted, "i''\in\mathfrak{C}\cap\mathfrak{W}"'] \arrow[dd, dotted, "i'\in\mathfrak{W}"] & A \arrow[dd, "i\in\mathfrak{C}"] \\
& X \arrow[dr, dotted, "p''\in\mathfrak{F}\cap\mathfrak{W}"'] & & \\
K \arrow[ur, dashed, "j"] \arrow[rr, dotted, "u"'] & & Z\times_B K \arrow[r, dashed, "v"'] & K
\end{tikzcd}$$

As $(v\circ p'')\circ j = id_K$, we can conclude that i is a retract of i''. But we already know that $\mathfrak{C}\cap\mathfrak{W}$ is closed under retracts, so $i\in\mathfrak{W}$, and therefore $f = p\circ i\in\mathfrak{W}$ as required.

To complete the proof, we have to show that $(N_1)-(N_5)$ imply (M_1) (which is trivial) and (M_2) (which we do now). Because of duality, it is enough to prove that $(\mathfrak{C}\cap\mathfrak{W},\mathfrak{F})$ is a weak factorization system. The factorization property (W_1) is just the same as (N_5), while the lifting property (W_2) is an immediate consequence of (N_4). Only the maximality (W_3) remains to be shown: Assume the morphism f has the right lifting property with respect to every morphism in $\mathfrak{C}\cap\mathfrak{W}$. By (M_5), we have a factorization $f = p\circ i$, where i is an acyclic cofibration and p a cofibration. According to our assumption, the diagonal filler in the following diagram exists:

$$\begin{tikzcd}
A \arrow[r, "id_A"] \arrow[d, "i"'] & A \arrow[d, "f"] \\
X \arrow[r, "p"'] \arrow[ur, dashed, "j"] & B
\end{tikzcd}$$

The following diagram proves that f is a retract of p.

$$\begin{tikzcd}
A \arrow[r, "i"] \arrow[d, "f"'] & X \arrow[r, "j"] \arrow[d, "p"'] & A \arrow[d, "f"] \\
B \arrow[r, "id_B"'] & B \arrow[r, "id_B"'] & B
\end{tikzcd}$$

By (N_3), we conclude $f\in\mathfrak{F}$, proving one part of the maximality condition (W_3). The other part follows by duality. $\square$

7 Groupoids

The Groupoid model is the easiest one of those we want to present here. At the same time, it is the oldest one ([21]) and probably the one that leaves us the least choices.

7.1 Weak factorization system on Grp

The basic setting is the category Grp with (small) groupoids as objects and functors as morphisms (so groupoids are the contexts). In category theory, there exist the so-called *Grothendieck fibrations* [37] and these are the fibrations we need, as described in [5], though it might be more appropriate to talk about *isofibrations*. As both concepts are the same in the simple case of groupoids, we do not make the distinction. We only give the definition for groupoids, not the more general one for categories.

Definition 7.1 (Fibrations in Grp)**.** A functor $p : E \to B$ between groupoids is a *fibration* if for any $e \in E$ and $f : b \to p(e)$, there is a morphism $g : e' \to e$ with $p(g) = f$. This immediately implies that every groupoid is fibrant.

The definition implies that any connected component of B is either disjoint from the image of p or completely objectwise contained in this image.

Groupoids carry the structure of a weak factorization system in the following way:

- $\mathfrak{L}$ is the class of functors which are injective and equivalences in the categorical meaning (i. e. embeddings which are injective on objects).
- $\mathfrak{R}$ are the fibrations defined above.

Theorem 7.2. *The structure* $(\mathfrak{C}, \mathfrak{F})$ *is a weak factorization system on the category of small groupoids.*

Proof. In my original composition, I had a very long and unnecessarily complicated proof here. I have decided to skip it, as there are much more elegant proofs available, for example at the *n*lab [13]. □

We may define A^I to be the arrow groupoid $A^{\to}$. Then, *refl* maps objects $a \in |A|$ canonically on $id_a \in |A^{\to}|$ and morphisms $m : a \to b$ on (m, m). It is easy to check that this is indeed an injective equivalence between A and $A^{\to}$, i.e. a trivial cofibration. Moreover, there is an obvious fibration $A^{\to} \to A \times_{\Gamma} A$ that represents the Identity Type. Summarised, our decomposition of the diagonal is

$$A \xrightarrow{a \mapsto id_a} A^{\to} \xrightarrow{p \mapsto (dom(p), codom(p))} A \times A$$

This model has in some way marked the beginning of the whole development. It is due to Martin Hofmann and Thomas Streicher's work [21], who used it to answer an important question about the *uniqueness of identity proofs*:

Corollary 7.3 (*UIP* is not provable from J, M. Hofmann and T. Streicher)**.** *Given two terms $a, b : A$ and two proofs $p, q : Id_A\, a\, b$, it is not possible to construct an inhabitant of $Id_{Id_A\, a\, b}\, p\, q$ without using axioms beyond J.*

Proof. We use the weak factorization system given above, together with theorem 5.5. We do not work out the details here. A subtle point is the question if it is possible to model dependent function spaces, as the category of small groupoids is not locally cartesian closed. In fact, it is, as shown by Hofmann & Streicher [21], and Palmgren [40] explains this by discussing that pullbacks along fibrations have "semi-strict pseudo-adjunctions".

Remark. It causes regularly some confusion that two objects in A^I such as $a \xrightarrow{f} b$ and $a \xrightarrow{g} b$, are always propositionally equal, since $A^I = A^{\rightarrow}$ and the diagram

$$\begin{tikzcd}
a \arrow[r, "f"] \arrow[d, dashed, "id_a"'] & b \arrow[d, dashed, "g \circ f^{-1}"] \\
a \arrow[r, "g"] & b
\end{tikzcd}$$

is obviously commutative. This should, however, not be too surprising, as [21] already mentioned that *UIP_tuple* is provable: Any (a, b, p) is equal to (a, a, refl_a), the crucial point is that p is *not* equal to refl_a.

Remark 7.4. While we have only used that small groupoids form a weak factorization system, they are even an example of a model category. To get this structure, take all the functors which are injective on objects as cofibrations and the usual categorical equivalences as weak equivalences.

8 Simplicial Sets

A *simplicial set* is a presheaf over the category of (isomorphy classes of) finite totally ordered nonempty sets and monotone functions. We want to introduce the most important concepts and provide some intuition.

8.1 General introduction to simplicial sets

Our main source for this subsection is Greg Friedman's article[15]. Another valuable article is [43] which is somewhat more direct but also discusses fewer concepts. As simplicial sets play an important role in homology and related topics, books such as [41], [17] and [32] can also serve as references. Here, we only give a brief summary. For everything beyond that, we highly recommend Friedman's article.

Definition 8.1 (category Δ)**.** For every natural number $n > 0$, we define $[n]$ to be the set $\{0, 1, \ldots, n-1\}$ (equivalently, the finite ordinal). Δ is the category that has the $[n]$ as objects and all monotone maps ($l \leq k$ implies $f(l) \leq f(k)$) as morphisms.

Remark 8.2. Caveat: The literature does not completely agree on the definition of Δ, but the different definitions are equivalent. It is still necessary to pay attention to avoid confusion. In particular, $[n]$ is sometimes defined to be $\{0, \ldots, n\}$ and the condition $n > 0$ is dropped, thereby shifting everything by 1. However, the empty set is usually never seen as an object of the category, i. e. Δ does not have an initial object.

Definition 8.3 (category **sSet**)**. sSet** is the functor category $\mathbf{Set}^{\Delta^{op}}$.

Although the above definition is short and precise, it is sometimes helpful to use a picture:

Given a simplicial set, i.e. a functor $X : \Delta^{op} \to \mathbf{Set}$, one can (to some extend) visualise $X[1]$ as a discrete set of points in the space. $X[2]$ is a set as well. We can visualise it as a set of directed connections, or lines, between pairs of points in $F[1]$. To see how this can be justified, notice that in Δ, there are two maps from the one-point set [1] to the two-point set [2]. Consequently, in Δ^{op}, there are two maps from [2] to [1]. Applying X on them, these are exactly the two maps that map every directed connection on its source respectively its target. Further, in Δ, there is exactly one map from [2] to [1]. After applying X, this map maps every point $x \in X[1]$ on the trivial connection from x to x. Therefore, the visualisation of only $X[1]$ and $X[2]$ looks like a directed multigraph with loops. Similarly, an object of the set $X[3]$ will be the shape of a triangle whose border is already given in the graph, $X[4]$ can be visualised as a tetrahedron, and so on. In fact, the image we have described can be seen as the *Grothendieck construction* $\int X$ (see)

In general, an element of $X[n]$ is called an *n-simplex of* X. We also call $X[n]$ the set of n-simplices. By the Yoneda lemma, this set can also be classified as $\mathbf{Set}^{\Delta^{op}}(\Delta^n, X)$.

It is handy to introduce a set of generators of the morphisms in Δ:

Definition 8.4. For i, n with $i \leq n$, we write D_i for the map $[n] \to [n+1]$ that is defined by

$$D_i(j) = \begin{cases} j & \text{if } j < i \\ j+1 & \text{otherwise.} \end{cases}$$

Further, again for $i < n$, there is the map $S_i : [n+1] \to [n]$, defined by

$$S_i(j) = \begin{cases} j & \text{if } j \leq i \\ j-1 & \text{otherwise.} \end{cases}$$

It is easy to see that those two classes of maps generate all the morphisms in Δ. They are central in the theory of simplicial sets:

Definition 8.5 (face and degeneracy maps)**.** Given a simplicial set X, the morphism $d_i := X(D_i) : X[n+1] \to X[n]$ is called *face map*, while $s_i := X(S_i) : [n] \to [n+1]$ is called *degeneracy map*.

The reason for these names can again be explained using the visualisation: d_i maps an element of $X[n+1]$, i.e. an $n+1$-simplex, on its ith face. This can also be expressed by saying that the ith corner is deleted, which collapses the rest. For $n = 1$, we have already seen that this morphism maps a directed line on one of its endpoints. For $n = 2$, it maps a triangle on one of its three faces, for $n = 3$, a tetrahedron is mapped on one of its four faces, and so on. Dually, s_i maps an n-simplex on an $n+1$-simplex by just using the ith corner twice. In the case of $n = 1$, we have described s_0 above as the map that maps a 1-simplex on the trivial connection to itself, i.e. a point is mapped on the degenerated line which has the point as both endpoints.

An n-simplex is therefore called *degenerated* if it can be written as $s_i(x)$ for some $n-1$-simplex x, else it is called *non-degenerated*.

A morphism in **sSet** is, of course, just a natural transformation between functors $F, G : \Delta^{op} \to \mathbf{Set}$. It maps points on points, lines on lines, triangles on triangles and so on and is therefore easy visualise as well. If we specify $\mu_{[n]}$ for such a natural transformation, all $\mu_{[m]}$ with $m < n$ are determined.

There is one type of simplicial sets that can, in some sense, be seen as the most basic type, often referred to as the *standard simplices*:

Definition 8.6 (standard simplex Δ^n)**.** For any positive integer n, we define $\Delta^n := y[n]$, where y is the *Yoneda embedding* $y : C \to \mathbf{Set}^{C^{op}}$ for a locally small category C. If we spell this out, Δ^n is the simplicial set given by the functor $\Delta(\cdot, [n]) : \Delta^{op} \to \mathbf{Set}$. Its visualised version looks like a (regular) triangle of dimension $n-1$; i.e., Δ^1 looks like a single point, Δ^2 like a line, Δ^3 like a triangle (with its body), Δ^4 like a tetrahedron, and so on.

Note that Δ^n always has exactly one non-degenerated n-simplex. More general, for each m, Δ^n contains exactly $\binom{n}{m}$ non-degenerated elements as any set of m distinct points (elements of $\Delta^n[1]$) form the boundary of exactly one non-degenerated m-simplex. Also note that Δ^1 is the terminal object of **sSet**. Caveat, again: With the alternative formalization of remark 8.2, Δ^0 is the terminal object. Another possible way of dealing with this is defining Δ^n to be $y[n+1]$.

8.2 Kan fibrations

Basically, a *Kan fibration* is a simplicial map satisfying a certain lifting property, which should not be surprising. Again, we recommend [15] and the articles mentioned above as our main references.

Definition 8.7 (k^{th} *horn* Λ^n_k)**.** For $k < n$, the k^{th} *horn* (denoted by Λ^n_k) of the simplex Δ^n can be defined by the full subcategory that is given by

$$\Lambda^n_k[j] := \{f : [j] \to [n] | \exists i \in [n]. i \neq k \wedge i \notin f[j]\}.$$

Here, we make use of the Yoneda embedding again. Λ^n_k is obtained by removing the "interior" and the $n-1$-dimensional boundary piece at position k. There is therefore an obvious inclusion $\Lambda^n_k \hookrightarrow \Delta^n$.

Definition 8.8 (Kan fibration)**.** Finally, a morphism $f : E \to B$ in **sSet** is a *Kan fibration* if, for any (k, n), any commutative diagram of the form

$$\begin{tikzcd}
\Lambda^n_k \arrow[r] \arrow[d, hook] & E \arrow[d, "f"] \\
\Delta^n \arrow[r] & B
\end{tikzcd}$$

has a diagonal filler $j : \Delta^n \to E$.

The idea is the same as for all fibrations: "If we can complete something in B, then we can also complete it in E". The fibrant objects, i.e. those objects E such that $E \to \Delta^1$ is a Kan fibration, are called *Kan complexes.*

Note that even such simple examples as Δ^n fail to be Kan complexes (see [15]):

$$\begin{tikzcd}
\Lambda^3_0 \arrow[r, "0\mapsto0{,}1\mapsto1{,}2\mapsto0"] \arrow[d, hook] & \Delta^2 \arrow[d, "1"] \\
\Delta^n \arrow[r] & \Delta^1
\end{tikzcd}$$

Λ^3_0 has the three constant functions as 1-simplices, which we have just called $0, 1, 2$ above. It is a good idea to think about the triangle with vertices labelled $0, 1, 2$ and, as it is the 0^{th} horn, without the body and without the edge between 1 and 2. The given mapping for these constants determines the whole simplicial map. Δ^2 is, thinking this way, the line with two endpoints, labelled 0 and 1. The upper morphism works well; however, there is *no* diagonal filler because we would have to map the edge between 1 and 2 to the edge between 1 and 0, but in the wrong way ($1 \mapsto 1, 2 \mapsto 0$), which is not a morphism in **sSet**.

8.3 Simplicial sets and spaces

Simplicial sets are used as a completely algebraic model of "nice" topological spaces. To make the connection clear, we first need to become more serious of what we have called "visualisation so far:

Definition 8.9 (Realisation of standard simplices)**.** For all $n > 0$, we denote by $|\Delta^n|$ the realisation of the standard simplex, that is the topological space given by

$$|\Delta^n| := \{(x_0, x_1, \ldots, x_n \in \mathbb{R}^{n+1} | 0 \leq x_i \leq 1, \sum x_i = 1\}$$

This definition is straightforward and well-known. We can use it to define a functor **Top** $\to$ **sSet**:

Definition 8.10 (singular set functor)**.** The *singular set functor* $\mathfrak{S}$: **Top** $\to$ **sSet** is given by

$$\mathfrak{S}Y := \mathbf{Top}(|\Delta^n|, Y)$$

This means, if Y is a space, then $\mathfrak{S}Y$ is the set of "ways how Δ^n can be embedded in Y", i.e. the set of all "pictures" of Δ^n. Of course, $\mathfrak{S}Y$ is very large unless Y is discrete.

We now want to be more precise about the notion of realisation, or "visualisation". While intuitively easy, it is surprisingly hard to define a functor that builds a space out of a simplicial set X in a reasonable way. The definition we state is given in [43]. We choose it as it is compact and at the same time not (much) more confusing than the "more down-to-earth" definition given in [15]. Clearly, any set can be viewed as a discrete space, in particular, $X[n]$ is one. Consider the product of spaces $|\Delta^m| \times X[n]$. Given $f : [n] \to [m]$ in Δ, there is a canonical continuous map $f_* : |\Delta^m| \times X[n] \to |\Delta^n| \times X[n]$ doing nothing on $X[n]$ and sending the standard simplex of dimension m to the one of dimension n. Similarly, there is the map $f^* : |\Delta^m| \times X[n] \to |\Delta^m| \times X[m]$. We now define (where we write $\coprod$ instead of $+$ for the coproduct):

Definition 8.11 (geometric realisation functor)**.** The *geometric realisation functor* $| \cdot |$: **sSet** $\to$ **Top** is given by

$$|X| := \mathit{colimit} \left(\coprod_{f:[n]\to[m]} |\Delta^m| \times X[n] \overset{f_*, f^*}{\Longrightarrow} \coprod_{[n]} |\Delta^n| \times X[n] \right)$$

Note that this is often always written as $\int^n |\Delta^n| \times X[n]$.[8]

[8]This is a generalized version of the Grothendieck construction (if I am not mistaken)

Intuitively, the realisation functor just does what it is supposed to do: For every standard simplex occurring in the simplicial set, it constructs its geometric version. This gives us quite a lot of "pyramids" in every dimension and we have to make sure that all the face and degeneracy maps are respected. This is done by taking the colimit.

Theorem 8.12 ($|\cdot| \dashv \mathfrak{S}$)**.** *The geometric realisation functor is left adjoint to the singular set functor.*

Proof. For an even more general statement see [43]. $\square$

Especially interesting is that Kan complexes are actually in some way the same as CW-complexes:

Theorem 8.13. *The category of Kan complexes and homotopy classes of maps between them is equivalent to the category of CW complexes and homotopy classes of continuous maps, where the equivalences are given by* $|\cdot|$ *and* $\mathfrak{S}$*.*

Proof. See [41], theorem I.11.4. $\square$

In Section 2, we have tried to give intuition of what homotopy type theory is about. While the ideas work nicely in our description, the category **Top** is not very well-behaved, which raises a lot of problems when it comes to the details of an interpretation of type theory. On the other hand, **sSet** is a purely combinatorial formulation with much better properties. The above theorem demonstrates how one should think, in summary: Work in **sSet**, but get intuition from **Top**!

8.4 The model in sSet

Multiple sources (including [5], [6]) explain that there is the following model structure on **sSet**:

1. cofibrations are the monomorphisms
2. weak equivalences are the weak homotopy equivalences (see below)
3. fibrations are the Kan fibrations

Therefore, the weak factorization system we should use is

- $\mathfrak{L} =$ *monos* $\cap$ *weak equivalences*
- $\mathfrak{R} =$ *Kan fibrations*

A morphism $f : X \to Y$ in **sSet** is a *weak homotopy equivalence* iff it induces isomorphisms on all homotopy groups. For the homotopy groups of a simplicial set X, there are several equivalent definitions (see [15], section 9), one of them saying that the n^{th} homotopy group is defined to be the n^{th} homotopy group of the topological space that is obtained by applying the realisation functor on X. Applying Whitehead's theorem [52], we should be able to conclude that a *weak homotopy equivalence* is just a map that becomes, after realization, a homotopy equivalence.

Simplicial sets are used by Voevodsky to model univalent type theory ([49], [50], [51]). For a good explanation of the construction (which is unfortunately quite involved), we would like to recommend Kapulkin & Lumsdaine & Voevodsky's [22] or Streicher's [47].

9 Univalence

We now switch to a different aspect: Instead of discussing model constructions, we examine interesting (possible) properties of identity types.

9.1 Contractibility

Contractibility is, in topology, a well-known property of topological spaces: A space is called contractible if (and only if) it is homotopically equivalent to the point. This means, a space X is contractible iff there exists a continuous map $H : X \times [0,1] \to X$ and a point $a \in X$ so that, for all $x \in X$, we have $F(x,0) = x$ and $F(x,1) = a$; in other words, F is at "time" 0 the identity and at "time" 1 constant.

For a type A, the notion is defined analogously:

Definition 9.1 (Contractibility)**.** A type A is called *contractible* if the type $\mathit{Contractible}\,(A) := \Sigma_{a:A}.\Pi_{a':A}.\mathit{Id}_A\,a\,a'$ is inhabited.

Unsurprisingly, this definition requires A to be inhabited by a distinguished element a. Furthermore, every other element has to be equal to a. At first, this property might look a bit weak: The corresponding ω-groupoid of A has, obviously, "up to propositional equality" exactly one 0-cell, but what about higher cells? There is no need to worry, as we will soon understand that this definition does indeed imply the same thing for all levels.

In the homotopy interpretation, the above definition looks like the definition of path-connectedness. However, if we have a closer look, we notice that it gets interpreted as *There exists a point $a \in A$ and a continuous function f which maps every point $x \in A$ on a path between a and x.* The continuity of f is the important detail. For example, consider the space S^1 It can, for example, be defined as the set of all points in the euclidean plane that have distance 1 from the origin. Another possible definition is to define it as the CW-complex with one 0-cell, where we attach one 1-cell in the obvious way; and this meets the type theoretic definition of the circle as a *higher inductive type* (see [29], [46]) quite well. For the moment, let us identify the circle with $[0, 2\pi]$, divided by the relation that unifies 0 and 2π. Assume that there is a continuous map f mapping a point x of this interval to a path from 0 to this point. $f(0)$ is a path from 0 to 0. Now, increase the argument x continuously; this makes the path $f(x)$ change continuously. Therefore, $f(2\pi)$ is homotopic to the path from 0 to 2π, composed with $f(0)$; but of course, $f(2\pi)$ is just $f(0)$, which shows that any path from 0 to itself is null-homotopic, contradicting the properties of the circle.

9.2 Homotopy Levels

The notion of *homotopy levels* corresponds (roughly) to the question which homotopy groups of a space are nontrivial. Clearly, for a contractible space, they are all trivial; and in fact, we define *H-level*$_0$ just to be the same as contractible. A space is still "relatively simple" if it becomes contractible after replacing it by it's path space (or iterating this step several times). For types, we define analogously:

Definition 9.2 (homotopy level). A type A is said to be of *homotopy level 0* just if it is contractible:

$$\textit{H-level}_0\,(A) := \textit{Contractible}\,(A)$$

Moreover, if all the identity types are of *homotopy level* n, then A is of *homotopy level* $n+1$.

$$\textit{H-level}_{n+1}\,(A) := \Pi_{ab:A}.\textit{H-level}_n\,(Id_A\,a\,b)$$

Remark 9.3. For small homotopy levels, the following notions are commonly used:

- A type of *homotopy level 0* is a singleton type, (isomorphic to) the unit type, or just contractible.
- A proposition, i.e. a type with at most one inhabitant, has *homotopy level 1.*
- The types of *homotopy level 2* are called *sets.*
- Less frequently used, but reasonable, is writing *groupoids* for types of *homotopy level 3.*

- Similarly, the *homotopy level 4* types are 2-groupoids, those of level 5 are 3-groupoids, and so on. In general, types do not need to have a finite homotopy level or the homotopy level might just not be provable. Univalence ensures that there is a type of homotopy level 3, the universe type. The hierarchy of universes determines for which levels a type exists, such that the type is provable not of the corresponding level.

9.3 Weak Equivalences

To understand the notion of a *weak equivalence*, homotopical intuition is, again, quite helpful. First of all, if $f : A \to B$ is a function and $b \in B$, we define the preimage of b:

Definition 9.4 (Preimage of $f : A \to B$)**.** The preimage of a function at b is defined as the set of pairs of a point, together with a proof that this point is indeed mapped to b:

$$f^{-1}b := \Sigma_{a:A}.\mathit{Id}_B\, b\, f(a)$$

Definition 9.5 (Weak equivalence property)**.** A function $f : A \to B$ is called a *weak equivalence* if all preimages are contractible:

$$\mathit{isWeq}\, f := \Pi_{b:B}.\mathit{Contractible}\left(f^{-1}b\right)$$

Definition 9.6 (Weak equivalence)**.** We define a *weak equivalence* between types A, B to be a map, together with a proof that this map is indeed a weak equivalence:

$$\mathit{Weq}\, A\, B := \Sigma_{f:A\to B}.\mathit{isWeq}\, f$$

Definition 9.7 (Weak equivalence, alternative)**.** An alternative definition for a *weak equivalence* between types A, B is a tuple consisting of a map in each direction, together with a proof that each composition is (extensionally equal to) the identity:

$$\mathit{WeqAlt}\, A\, B := \Sigma_{\phi:A\to B,\psi:B\to A}.\Pi_{a:A}.\mathit{Id}_A\,(\psi\circ\phi(a))\, a \times \Pi_{b:B}.\mathit{Id}_B\,(\phi\circ\psi(b))\, b$$

Remark 9.8. The two notions of weak equivalence are logically equivalent, but not isomorphic. However, it is a (proven?) conjecture that we can make them isomorphic by adding a concrete coherence proof to the second one, thereby making the 4-tuple a 5-tuple. The coherence condition follows from the other terms, but the crucial point is that no *unique* proof follows.

Lemma 9.9 (Composition with weak equivalences is weak equivalence)**.** *Assume A, B, C are types. If w : $\mathit{Weq}\; B\; C$ is a weak equivalence, then composition with w is a weak equivalence, i.e $\mathit{Weq}\,(A \to B)\,(A \to C)$ is inhabited.*

Proof. If u is the inverse of w in the alternative definition of a weak equivalence, it is enough to show that $\lambda f.w \circ f$ and $\lambda f.u \circ f$ are inverse. More precisely, it is sufficient to prove that their composition is extensionally equal to the identity on $A \to B$ respective $A \to C$, which is straightforward. $\square$

9.4 The Univalence Axiom

From the previous section, it is clear that the identity function on any type A is always a weak equivalence (more precisely, can be completed canonically to a weak equivalence); by *idIsWeq* , we denote the canonical map of type $\forall A\,.\,\mathit{Weq}\;\, A\,A$. Assume A, B are types. Furthermore, assume p is an inhabitant of $\mathit{Id}\,A\,B$. From p, we can construct a weak equivalence between A and B: Using the J eliminator, we only have to give this construction if p is the reflexivity proof; but in that case, *idIsWeq* is just what we need. The *Univalence Axiom* states that this map is again a weak equivalence.

The contents of this sections are, by the best of my knowledge, originally by Voevodsky; they are nicely presented in Bauer & Lumsdaine's notes [28].

We first define the "canonical map" precisely:

Definition 9.10. There is map of type $\forall A\,B\,.\,\mathit{Id}\,A\,B \to \mathit{Weq}\;\, A\,B$, constructed as

$$\mathit{eqToWeq} \;= J\,(\lambda\,A\,B\,.\,(p : \mathit{Id}\,A\,B) \to \mathit{Weq}\;\, A\,B)\,\mathit{idIsWeq}$$

Definition 9.11 (Univalence Axiom)**.** The map *eqToWeq* is a weak equivalence. In other words, the Univalence axiom postulates a term of type

$$\forall A\,B\,.\,\mathit{isWeq}\,\mathit{eqToWeq}$$

The Univalence Axiom provides us with the possibility to treat weak equivalences similarly as propositional equalities. To make this clear, we prove the following:

Theorem 9.12 (Induction on weak equivalences)**.** *Given some*

$$P : \forall\, U, V\,.\,\mathit{Weq}\,U\;V \to \mathit{Type},$$

assume we can construct a term for "canonical weak equivalences". More precisely, assume we can construct a term of the type

$$m : \forall T\,.\,P\,T\,T\,(\mathit{idIsWeq}\;T)$$

Then we can also construct an inhabitant of P.

Proof. Define

$$P' : \forall U\, V \,.\, \mathit{Id} U\, V \to \mathit{Type}$$

by

$$P' = \lambda U\, V\, q \,.\, P\, U\, V\, (\mathit{eqToWeq}\ q)$$

Now, $\forall U \,.\, P'\, U\, U\, \mathit{refl}_U$ is inhabited by m (this uses the β rule of identity types). By J, P' is inhabited. Given any U, V as well as $w : \mathit{Weq}\ \ U\, V$ and univalence, we get a proof $p : \mathit{Id} U\, V$. But $\mathit{eqToWeq}\, p$ is equal to w, so using the constructed inhabitant of $P'\, U\, V\, p$ and J (or just a substitution rule that follows from J), we get an inhabitant of $P\, U\, V\, w$. □

We want to conclude with a proof that Univalence implies Extensionality of functions, i.e., if two functions are pointwise equal, we can prove that they are equal. We summarise the main argument of [28].

Lemma 9.13 (source and targent are Weak Equivalences)**.** *Recall that we write A^I for $\Sigma_{a,b:A}.\mathit{Id}_A\, a\, b$. Given a type A, the canonical projection maps $\mathit{src}_A : A^I \to A$ and $\mathit{trg}_A : A^I \to A$ are weak equivalences.*

Proof. We only give a sketch of the proof for the *src* function. Here, it seems to be advantageous to use definition 9.7. We want to prove that the map $r_A : A \to A^I$ is an inverse of src_A (recall $r_A = \lambda a \,.\, (a, a, \mathit{refl}_a)$. It is clear that $\mathit{src}_A \circ r_A$ is extensionally equal to id_A. For the other direction, we have to show that every term $(a, b, p) : A^I$ equals (a, a, refl_a). But, using the J eliminator, it is enough to show this if (a, b, p) is (a, a, refl_a), and in this case, it follows by reflexivity. □

Theorem 9.14 (Univalence and η imply Extensionality)**.** *Assume we have, for types A, B and functions $f, g : A \to B$, a proof that f and g are pointwise equal; i.e. we have $p : \Pi_{a:A}.\mathit{Id}_A\, (fa)\, (ga)$. Using the Univalence axiom (and the usual η law for functions), we can construct an inhabitant of $\mathit{Id}_{A\to B}\, f\, g$.*

Proof. We sketch the proof that is given in [28]. Define

$$d := \lambda a \,.\, (fa, fa, \mathit{refl}_{fa})$$
$$e := \lambda a \,.\, (fa, ga, pa)$$

Now, $\mathit{src}_A \circ d = \lambda a \,.\, fa = \mathit{src}_A \circ e$. But for any weak equivalence s, $\mathit{Id}\, d\, e$ is inhabited iff $\mathit{Id}(\, s \circ d)(s \circ e)$ is, which is easily shown by induction on weak equivalences. We therefore just need to apply lemma 9.12 to see that $\mathit{Id}\, d\, e$ is inhabited and also $\mathit{Id}\, (\mathit{trg}_A \circ d)\, (\mathit{trg}_A \circ e)$, which is just $\mathit{Id}\, (\lambda a \,.\, fa)\, (\lambda a \,.\, ga)$, so the η law solves it. □

Remark 9.15. For simplicity, we have only stated the nondependend form of extensionality. The dependent version holds as well, but is more involved.

10 Hedberg's Theorem

In 1998, Michael Hedberg has published a proof that, for a given type, decidable equality implies uniqueness of identity proofs [18]. His original proof is quite lengthy, though it provides a couple of very interesting insights. Here, I want to present a much more direct proof, which I have also formalised in Coq (available on my homepage). There is also a post on the HoTT blog [1] on the topic.

Definition 10.1 (decidability)**.** A type A is said to be *decidable* if there is either a proof that it is inhabited or a proof that it is not:

$$Decidable_A = A + \neg A$$

where, of course, $\neg A$ is just short-hand for $A \to \bot$. *Decidable equality* means that we can, for each pair of terms, decide their equality type:

$$DecEqu_A = \forall a\, b\, .\, Decidable_{(Id_A\, a\, b)}$$

Uniqueness of identity proofs has already been introduced at the very beginning of this composition, we just repeat the definition in the form of a type:

Definition 10.2 (uip)**.**

$$UIP_A = \forall a\, b : A\, .\, \forall p\, q\ :\ Id_A\, p\, q\, .\, Id\, p\, q$$

Hedberg's theorem states that decidable equality implies UIP:

Theorem 10.3 (Hedberg)**.**

$$DecEqu_A \to UIP_A$$

Proof. Assume *dec: deceqA* (in some context Γ). Further, assume (a, b, p) : A^I (in the context). We can now "ask" the "deciding function" *dec* what it "thinks" about a, b respectively a, a; it will either tell us that they are equal or unequal. The latter case would, however, immediately lead to a contradiction, as we already know that a and b are equal. Therefore,

$$\begin{aligned} dec\, a\, b &= \texttt{inl}\ \ q_1 \qquad \textit{for some } q_1 : Id_A\, a\, b \\ dec\, a\, a &= \texttt{inl}\ \ q_2 \qquad \textit{for some } q_2 : Id_A\, a\, a \end{aligned}$$

We claim that p equals $q_1 \circ q_2^{-1}$ propositionally (using the notation of remark 2.2). But applying J, we only need to prove it for $(a, b, p) = (a, a, \mathit{refl}_a)$, in which case q_1 and q_2 are the same, so that it suffices to observe that $q_2 \circ q_2^{-1}$ equals reflexivity. As every inhabitant of $Id_A\, a\, b$ equals $q_1 \circ q_2^{-1}$, there cannot be more than one. □

References

[1] *A direct proof of Hedberg's theorem.* URL: http://homotopytypetheory.org/2012/03/30/a-direct-proof-of-hedbergs-theorem/.

[2] Andreas Abel and Ulrich Schöpp. *Semantik von Programmiersprachen.*

[3] Peter Arndt and Krzysztof Kapulkin. "Homotopy-theoretic models of type theory". In: *Proceedings of the 10th international conference on Typed lambda calculi and applications.* TLCA'11. Novi Sad, Serbia: Springer-Verlag, 2011, pp. 45–60. ISBN: 978-3-642-21690-9.

[4] Steve Awodey. *Category Theory.* 2006. ISBN: 0198568614. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0198568614.

[5] Steve Awodey. *Type Theory and Homotopy.* Tech. rep. arXiv:1010.1810v1. URL: www.andrew.cmu.edu/user/awodey/preprints/TTH.pdf.

[6] Steve Awodey and Michael A Warren. *Homotopy theoretic models of identity types.* Tech. rep. arXiv:0709.0248. 2007. URL: http://arxiv.org/abs/arxiv:0709.0248.

[7] Benno van den Berg and Richard Garner. *Topological and simplicial models of identity types.* Tech. rep. arXiv:1007.4638. 2010. 47 pp. URL: http://arxiv.org/abs/1007.4638.

[8] Benno van den Berg and Richard Garner. *Types are weak omega-groupoids.* Tech. rep. 2008. URL: http://arxiv.org/abs/arXiv:0812.0298.

[9] Aldridge Bousfield. "Constructions of factorization systems in categories". In: *Journal of Pure and Applied Algebra 9 (1977)* (), pp. 207–220.

[10] P.-L. Curien. "Substitution up to isomorphism". In: *Fundam. Inf.* 19.1-2 (Sept. 1993), pp. 51–85. ISSN: 0169-2968. URL: http://dl.acm.org/citation.cfm?id=175469.175471.

[11] Nils Anders Danielsson. *Chalmers: Equality.* URL: http://www.cse.chalmers.se/~nad/listings/equality/README.html.

[12] Dwyer and Spalinski. "Homotopy Theories and Model Categories". In: (). URL: folk.uio.no/paularne/SUPh05.

[13] n*lab.* URL: http://ncatlab.org/nlab/show/HomePage.

[14] Steve Awodey et.al. *HOTT (Blog about homotopy type theory).* URL: http://homotopytypetheory.org/.

[15] Greg Friedman. "An elementary illustrated introduction to simplicial sets". In: (2008). URL: http://arxiv.org/abs/0809.4221.

[16] Nicola Gambino and Richard Garner. “The identity type weak factorisation system”. In: *Theor. Comput. Sci.* 409 (1 2008), pp. 94–109. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2008.08.030. URL: http://www.comp.mq.edu.au/~rgarner/Idtype/Idtype.html.

[17] Allen Hatcher. *Algebraic Topology.* 1st ed. Cambridge University Press, Dec. 2001. ISBN: 0521795400. URL: http://www.math.cornell.edu/~hatcher/.

[18] Michael Hedberg. “A coherence theorem for Martin-Löf’s type theory”. In: *J. Functional Programming* (1998), pp. 4–8. URL: http://www.andrew.cmu.edu/user/awodey/hott/papers/hedberg.pdf.

[19] Martin Hofmann. “On the Interpretation of Type Theory in Locally Cartesian Closed Categories”. In: *Proceedings of Computer Science Logic, Lecture Notes in Computer Science.* Springer, 1994, pp. 427–441. URL: www.dcs.ed.ac.uk/home/mxh/csl94.ps.gz.

[20] Martin Hofmann. “Syntax and Semantics of Dependent Types”. In: *Semantics and Logics of Computation.* Cambridge University Press, 1997, pp. 79–130. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.8985&rep=rep1&type=pdf.

[21] Martin Hofmann and Thomas Streicher. “The Groupoid Interpretation of Type Theory”. In: *In Venice Festschrift.* Oxford University Press, 1996, pp. 83–111. URL: www.mathematik.tu-darmstadt.de/~streicher/venedig.ps.gz.

[22] Chris Kapulkin, Peter LeFanu Lumsdaine, and Vladimir Voevodsky. “Univalence in Simplicial Sets”. In: (). URL: http://arxiv.org/abs/1203.2553v2.

[23] Krzysztof Kapulkin. *Homotopy-theoretic models of type theory (master thesis).*

[24] Thomas Köppe. *Basic homotopy theory.* URL: www.maths.ed.ac.uk/~s0571100/Homotopy.pdf.

[25] Nicolai Kraus. *On Hedberg’s theorem: Proving and Painting.* talk. URL: red.cs.nott.ac.uk/~ngk/away2012_nicolai.pdf.

[26] J. Lambek and P. J. Scott. *Introduction to higher order categorical logic.* New York, NY, USA: Cambridge University Press, 1988. ISBN: 0-521-35653-9.

[27] Dan Licata. *Just Kidding: Understanding Identity Elimination in Homotopy Type Theory.* URL: http://homotopytypetheory.org/2011/04/10/just-kidding-understanding-identity-elimination-in-homotopy-type-theory/.

[28] Andrej Bauerand Peter LeFanu Lumsdaine. *A Coq proof that Univalence Axioms implies Functional Extensionality.* URL: https://github.com/andrejbauer/Homotopy/raw/master/OberwolfachTutorial/univalence.pdf.

[29] Peter LeFanu Lumsdaine. *Higher Inductive Types: a tour of the menagerie.* URL: http://homotopytypetheory.org/2011/04/24/higher-inductive-types-a-tour-of-the-menagerie.

[30] Peter LeFanu Lumsdaine. "Model Structures from Higher Inductive Types". In: (). URL: http://www.mathstat.dal.ca/~p.l.lumsdaine/research/Lumsdaine-Model-strux-from-HITs.pdf.

[31] Peter Lefanu Lumsdaine. "Weak omega-Categories from Intensional Type Theory". In: *Proceedings of the 9th International Conference on Typed Lambda Calculi and Applications.* TLCA '09. Brasilia, Brazil: Springer-Verlag, 2009, pp. 172–187. ISBN: 978-3-642-02272-2. URL: http://dx.doi.org/10.1007/978-3-642-02273-9_14.

[32] Mark Hovey. "Model Categories". 2000. URL: http://www.math.tamu.edu/~plfilho/wk-seminar/Hovey_book.ps.

[33] Per Martin-Löf. *An Intuitionistic Theory of Types.* Tech. rep. 1972.

[34] Per Martin-Löf. "An Intuitionistic Theory of Types: Predicative Part." In: *The Journal of Symbolic Logic* 49.1 (Mar. 1984), pp. 311+. ISSN: 00224812. DOI: 10.2307/2274116. URL: http://dx.doi.org/10.2307/2274116.

[35] Per Martin-Löf. *Intuitionistic type theory.* Vol. 1. Studies in Proof Theory. Naples: Bibliopolis, 1984. URL: http://www.ams.org/mathscinet-getitem?mr=769301.

[36] Conor McBride. "Dependently Typed Functional Programs and their Proofs". PhD thesis. University of Edinburgh, 1999. URL: http://www.lfcs.informatics.ed.ac.uk/reports/00/ECS-LFCS-00-419/.

[37] *ncatlab: Grothendieck fibration.* URL: http://ncatlab.org/nlab/show/Grothendieck+fibration.

[38] *ncatlab: Model Category.* URL: http://nlab.mathforge.org/nlab/show/model+category.

[39] *ncatlab: weak factorization system.* URL: http://ncatlab.org/nlab/show/weak+factorization+system.

[40] Erik Palmgren. *Groupoids and Local Cartesian Closure.* 2003. URL: www2.math.uu.se/~palmgren/gpdlcc.pdf.

[41] P.G. Goerss and J.F. Jardine. *Simplicial Homotopy Theory.* URL: http://www.math.uiuc.edu/K-theory/0148/.

[42] D. G. Quillen. *Homotopical Algebra.* Vol. 43. Lecture Notes in Mathematics. Springer-Verlag, 1967.

[43] Emily Riehl. *A leisurely introduction to simplicial sets.* URL: www.math.harvard.edu/~eriehl/ssets.pdf.

[44] Christian Sattler. "On the Merits of the η-Law for Inductive Types". 2012.

[45] R. A. G. Seely. "Locally Cartesian Closed Categories and Type Theory". In: *Math. Proc. Camb. Phil. Soc.* 95 (1984), pp. 33–48. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0198568614.

[46] Mike Shulman. *Homotopy Type Theory (several Blogposts).* URL: http://golem.ph.utexas.edu/category/2011/03/homotopy_type_theory_i.html.

[47] Thomas Streicher. "A Model of Type Theory in Simplicial Sets". In: (). URL: http://www.mathematik.tu-darmstadt.de/~streicher/.

[48] *The nLab.* URL: http://ncatlab.org.

[49] V. Voevodsky. "The equivalence axiom and univalent models of type theory". In: *Talk at CMU* (). URL: http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/CMU_talk.pdf.

[50] V. Voevodsky. "Univalent Foundations of Mathematics". In: *Talk in Goteborg* (). URL: www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/2011_Goteborg.pdf.

[51] V. Voevodsky. "Univalent Foundations Project". In: *a modified version of an NSF grant application* (). URL: www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/univalent_foundations_project.pdf.

[52] *Wikipedia: Whitehead theorem.* URL: http://en.wikipedia.org/wiki/Whitehead_theorem.