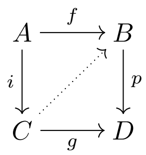
1 Factorisation axioms for type theory

Definition 1. Let $\mathcal{C}$ be a category, and let $i\colon A \to B$ and $p\colon C \to D$ be maps of $\mathcal{C}$. We say that i and p are an **extension-lifting pair**, and write $i \,\square\, p$, if, whenever we have a diagram of unbroken arrows



there exists a (not necessarily unique) fill-in as indicated by the broken arrow, making both triangles commute.

We are now going to utilise this notion to analyse intensional type theory. We work in polymorphic intensional type theory à la Martin-Löf, which we will denote by $\mathcal{T}$. Recall that we can generate a category $\mathcal{C}$ from such a type theory whose *objects* are contexts Γ modulo definitional equality and whose *morphisms* $\Gamma \to \Delta$ are context substitutions: if $\Delta = (y_1 : B_1, \ldots, y_m : B_m)$, then $f\colon \Gamma \to \Delta$ is given by judgements:

$$\begin{aligned}
&\Gamma \vdash f_1 \colon B_1\\
&\Gamma \vdash f_2 \colon B_2[f_1/y_1]\\
&\quad \cdots\\
&\Gamma \vdash f_m \colon B_m[f_1/y_1, \ldots, f_{m-1}/y_{m-1}]
\end{aligned}$$

modulo definitional equality in $\mathcal{T}$. We will write this as

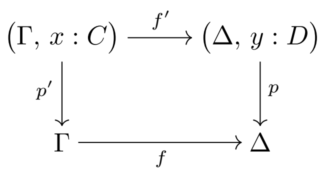
$$\mathbf{x} : \Gamma \vdash f(\mathbf{x}) : \Delta$$

for short. Moreover, every judgement of the form $\Gamma \vdash A\ \mathsf{type}$ induces a "dependent projection" map

$$(\Gamma,\, a : A) \to \Gamma$$

in $\mathcal{C}$, which simply projects away the last factor. We write $\mathcal{M}$ for the class of all morphisms of $\mathcal{C}$ of this form, and write $\mathcal{E}$ for the class ${}^{\square}\mathcal{M}$.

Proposition 2. *The class $\mathcal{M}$ is stable under pullback; that is, given a map $p\colon (\Delta,\, y : D) \to \Delta$ in $\mathcal{M}$ and an arbitrary map $F\colon \Gamma \to \Delta$, there is a pullback diagram of the form*



with $p' \in \mathcal{M}$.

Proof. Suppose that $\Delta = (y_1 : B_1, \ldots, y_m : B_m)$. We form the judgement

$$\mathbf{x} : \Gamma \vdash C(\mathbf{x}) := D[f(\mathbf{x})/\mathbf{y}]\ \mathsf{type},$$

take p' to be the corresponding dependent projection and take the map f' to be $(f,\, x)$. The universal property of pullback is easily verified. $\square$

We shall now examine some of the constructions of intensional type theory. Note that in what follows, we shall act as if there is no “ambient context”; to make things rigourous, we should really prepend everything we do with an extra context Θ, but for the sake of clarity we shall omit it. We look first at the intensional sum type, which obeys the following four equations:

$$\frac{A\ \mathsf{type} \qquad x : A \vdash B(x)\ \mathsf{type}}{\Sigma(A, B)\ \mathsf{type}} \tag{1}$$

$$\frac{a : A \qquad b : B(a)}{\langle a, b\rangle : \Sigma(A, B)} \tag{2}$$

$$\frac{y : \Sigma(A, B) \vdash C(y)\ \mathsf{type} \qquad a : A,\ b : B(a) \vdash d(a, b) : C(\langle a, b\rangle)}{y : \Sigma(A, B) \vdash J_d(y) : C(y)} \tag{3}$$

$$\frac{y : \Sigma(A, B) \vdash C\ \mathsf{type} \qquad a : A,\ b : B(a) \vdash d(a, b) : C(\langle a, b\rangle)}{a : A,\ b : B(a) \vdash J_d(\langle a, b\rangle) = d(a, b) : C(\langle a, b\rangle)} \tag{4}$$

What do these say when interpreted in $\mathcal{C}$? (1) merely asserts the existence of a certain type, whilst (2) asserts the existence of a context morphism

$$i\colon \big(a : A,\ b : B(a)\big) \to \big(y : \Sigma(A, B)\big).$$

We now turn to the interpretations of (3) and (4). Let us write $i \,\Box\, \mathcal{M}$ as an abbreviation for $i \,\Box\, p$ for all $p \in \mathcal{M}$.

Proposition 3. *Conditions* (3) *and* (4) *are equivalent to the statement that* $i \,\Box\, \mathcal{M}$.

Proof. Suppose first that $i \,\Box\, \mathcal{M}$, and that the hypotheses of (3) hold. From the judgement $y : \Sigma(A, B) \vdash C(y)\ \mathsf{type}$ we obtain as above a context morphism in $\mathcal{M}$ given by

$$p\colon \big(y : \Sigma(A, B),\ c : C(y)\big) \to \big(y : \Sigma(A, B)\big),$$

whilst $a : A,\ b : B(a) \vdash d(a, b) : C(\langle a, b\rangle)$ asserts the existence of a context morphism

$$d\colon \big(a : A,\ b : B(a)\big) \to \big(y : \Sigma(A, B),\ c : C(y)\big)$$

such that $p \circ d = i$; in other words, such that the following diagram of solid arrows commutes:

$$\begin{array}{ccc} \big(a : A,\ b : B(a)\big) & \xrightarrow{\ d\ } & \big(y : \Sigma(A, B),\ c : C(y)\big) \\ {\scriptstyle i}\big\downarrow & & \big\downarrow{\scriptstyle p} \\ \big(y : \Sigma(A, B)\big) & \xrightarrow[\mathrm{id}]{\qquad} & \big(y : \Sigma(A, B)\big). \end{array} \tag{5}$$

Now since $p \in \mathcal{M}$ we have $i \,\Box\, p$, and so there exists a diagonal fill-in

$$J_d\colon \big(y : \Sigma(A, B)\big) \to \big(y : \Sigma(A, B),\ c : C(y)\big)$$

for this square, making both triangles commute. Since the bottom triangle commutes, giving J is the same as giving a dependent element $y : \Sigma(A, B) \vdash$

$J_d(y) : C(y)$, which is what is required for the conclusion of (3); and since the top triangle commutes, the conclusion of (4) holds.

Conversely, if conditions (3) and (4) hold, then any diagram like (5) has a diagonal fill-in making both triangles commute. This is sufficient to show that $i \,\square\, p$ for any dependent projection $p \colon (\Gamma, d : D) \to \Gamma$. Indeed, suppose we are given a commutative square

$$\begin{array}{ccc} \big(a : A,\, b : B(a)\big) & \xrightarrow{f} & \big(\Gamma,\, d : D\big) \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p} \\ \big(y : \Sigma(A, B)\big) & \xrightarrow[g]{} & \Gamma. \end{array} \tag{6}$$

Then by Proposition 2, there is a pullback diagram of the form

$$\begin{array}{ccc} \big(y : \Sigma(A, B),\, c : C\big) & \xrightarrow{g'} & \big(\Gamma,\, d : D\big) \\ {\scriptstyle p'}\downarrow & & \downarrow{\scriptstyle p} \\ \big(y : \Sigma(A, B)\big) & \xrightarrow[g]{} & \Gamma \end{array}$$

with p' in $\mathcal{M}$, and so the diagram (6) induces a map $d \colon \big(a : A,\, b : B(a)\big) \to \big(y : \Sigma(A, B),\, c : C\big)$ satisfying $p'd = i$ and $g'd = f$. Now consider the diagram

$$\begin{array}{ccccc} \big(a : A,\, b : B(a)\big) & \xrightarrow{d} & \big(y : \Sigma(A, B),\, c : C)\big) & \xrightarrow{g'} & \big(\Gamma,\, d : D\big) \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p'} & & \downarrow{\scriptstyle p} \\ \big(y : \Sigma(A, B)\big) & \xrightarrow[\mathrm{id}]{} & \big(y : \Sigma(A, B)\big) & \xrightarrow[g]{} & \Gamma. \end{array}$$

The left hand square is a diagram like (5), and so has a fill-in $J_d \colon \big(y : \Sigma(A, B)\big) \to \big(y : \Sigma(A, B),\, c : C(y)\big)$ making both triangles commute. The outer rectangle is precisely diagram (6), and $g' \circ J_d$ is a fill-in for it making making both outer triangles commute. $\square$

Remark. The last part of the above proof can also be done inside the type theory; if we express the property that $i \,\square\, \mathcal{M}$ directly in type-theoretic notation, we end up with the following two conditions (where again we use "vector notation" for elements of Γ):

$$\frac{\mathbf{x} : \Gamma \vdash D \;\mathsf{type} \qquad y : \Sigma(A, B) \vdash g(y) : \Gamma \qquad a : A,\, b : B(a) \vdash f(a, b) : D(g(\langle a, b \rangle))}{y : \Sigma(A, B) \vdash J_f(y) : D(g(y))}$$

$$\frac{\mathbf{x} : \Gamma \vdash D \;\mathsf{type} \qquad y : \Sigma(A, B) \vdash g(y) : \Gamma \qquad a : A,\, b : B(a) \vdash f(a, b) : D(g(\langle a, b \rangle))}{y : \Sigma(A, B) \vdash J_f(\langle a, b \rangle) = f(a, b) : D(g(\langle a, b \rangle))}$$

and it is almost immediately obvious that these follow from conditions (3) and (4).

We turn now to the intensional identity type. This is given by judgements

$$\frac{A \text{ type}}{x : A,\ y : A \vdash \mathsf{Id}_A(x, y) \text{ type}} \tag{7}$$

$$\frac{a : A}{r(a) : \mathsf{Id}_A(a, a)} \tag{8}$$

$$\frac{x : A,\ y : A,\ z : \mathsf{Id}_A(x, y) \vdash C(x, y, z) \text{ type} \qquad a : A \vdash d(a) : C(a, a, r(a))}{x : A,\ y : A,\ z : \mathsf{Id}_A(x, y) \vdash J_d(x, y, z) : C(x, y, z)} \tag{9}$$

$$\frac{x : A,\ y : A,\ z : \mathsf{Id}_A(x, y) \vdash C \text{ type} \qquad a : A \vdash d(a) : C(a, a, r(a))}{a : A \vdash J_d(a, a, r(a)) = d(a) : C(a, a, r(a))}. \tag{10}$$

The pattern here is somewhat similar to before but a little more refined. Equation (7) asserts the existence of a map

$$p\colon \big(x : A,\ y : A,\ z : \mathsf{Id}_A(x, y)\big) \to \big(x : A,\ y : A\big)$$

in $\mathcal{M}$ whilst equation (8) asserts the existence of a map

$$i\colon \big(x : A\big) \to \big(x : A,\ y : A,\ z : \mathsf{Id}_A(x, y)\big)$$

such that the composite pi is the map $\Delta = (x, x)\colon \big(x : A\big) \to \big(x : A,\ y : A\big)$. The proof of the following is now identical in nature to the proof of Proposition 3:

Proposition 4. *Conditions* (9) *and* (10) *are equivalent to the statement that* $i \,\square\, \mathcal{M}$.

Thus we have shown the following:

- Given a judgement $a : A \vdash B(a)$ type, we can factor the unique context morphism into the terminal context $\big(a : A,\ b : B(a)\big) \to ()$ as

  $$\big(a : A,\ b : B(a)\big) \xrightarrow{i} \big(y : \Sigma(A, B)\big) \xrightarrow{p} (),$$

  where $p \in \mathcal{M}$ and $i \,\square\, \mathcal{M}$.

- Given a judgement $\vdash A$ type, we can factor the context map $\Delta\colon \big(x : A\big) \to \big(x : A,\ y : A\big)$ as

  $$\big(x : A\big) \xrightarrow{i} \big(x : A,\ y : A,\ z : \mathsf{Id}_A(x, y)\big) \xrightarrow{p} \big(x : A,\ y : A\big)$$

  where $p \in \mathcal{M}$ and $i \,\square\, \mathcal{M}$.

This suggests that we could replace the axioms for the identity and sum types with a new axiom scheme which we state in terms of $\mathcal{C}$:

Axiom. *Any context map* $F\colon \Gamma \to \Delta$ *can be factored as* $F = pi$ *where* $p \in \mathcal{M}$ *and* $i \,\square\, \mathcal{M}$.

This can be seen as a categorical counterpart to the most simplistic of Dybjer's "inductive schemata". If we translate it back into our type theory, we get intensional factorisation types:

$$\frac{\mathbf{x} : \Gamma \vdash f(\mathbf{x}) : \Delta}{\mathbf{y} : \Delta \vdash \Phi_f(\mathbf{y}) \ \mathsf{type}} \tag{11}$$

$$\frac{\mathbf{x} : \Gamma}{i(\mathbf{x}) : \Phi_f\big(f(\mathbf{x})\big)} \tag{12}$$

$$\frac{\mathbf{y} : \Delta,\, z : \Phi_f(\mathbf{y}) \vdash C(\mathbf{y}, z) \ \mathsf{type} \qquad \mathbf{x} : \Gamma \vdash d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big)}{\mathbf{y} : \Delta,\, z : \Phi_f(\mathbf{y}) \vdash J_d(\mathbf{y}, z) : C(\mathbf{y}, z)} \tag{13}$$

$$\frac{\mathbf{y} : \Delta,\, z : \Phi_f(\mathbf{y}) \vdash C(\mathbf{y}, z) \ \mathsf{type} \qquad \mathbf{x} : \Gamma \vdash d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big)}{\mathbf{y} : \Delta,\, z : \Phi_f(\mathbf{y}) \vdash J_d\big(f(\mathbf{x}), i(\mathbf{x})\big) = d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big).} \tag{14}$$

Immediately, factorisation types subsume the identity and sum type constructors; they also subsume the intensional unit type, which is the factorisation type for the unique context map $() \to ()$. What else does they let us do? Suppose that $\mathcal{T}$ also has the intensional boolean type $\mathbf{2}$, with two canonical elements 0 and 1; then we can define a map $f \colon () \to (y : \mathbf{2})$ picking out the canonical element $1 : \mathbf{2}$. The factorisation type of this map is a type $x : \mathbf{2} \vdash B(x)$ type satisfying the axioms

$$\frac{}{* : B(1)}$$

$$\frac{y : \mathbf{2},\, z : B(y) \vdash C(y, z) \ \mathsf{type} \qquad d : C(1, *)}{y : \mathbf{2},\, z : B(y) \vdash J_d(y, z) : C(y, z)}$$

$$\frac{y : \mathbf{2},\, z : B(y) \vdash C(y, z) \ \mathsf{type} \qquad d : C(1, *)}{y : \mathbf{2},\, z : B(y) \vdash J_d(1, *) = d : C(y, z).}$$

So informally, B is the "closure under propositional equality" of

$$B(0) = \emptyset \qquad \text{and} \qquad B(1) = \{*\}.$$

In general, our factorisation axiom, when applied to a context map $f \colon \Gamma \to \Delta$, yields a type Φ_f in context Δ which is inductively generated by the elements of Γ; the map f picks out which fibre of Φ_f over Δ each generating element of Γ will land in.

Remark. Our factorisation axiom is inspired by the categorical structure of a *weak factorisation system.* Given a category $\mathcal{C}$ and two classes $\mathcal{I}$ and $\mathcal{P}$ of its morphisms, we say that $(\mathcal{I}, \mathcal{P})$ is a **weak factorisation system** if the following two conditions are satisfied:

1. Every map $f \in \mathcal{C}$ can be factored as $f = pi$ with $i \in \mathcal{I}$ and $p \in \mathcal{P}$;
2. $\mathcal{I} = {}^{\square}\mathcal{P}$ and $\mathcal{P} = \mathcal{I}^{\square}$

where we write

$${}^{\square}\mathcal{F} := \{i \in \mathcal{C} : i \,\square\, \mathcal{F}\} \qquad \text{and} \qquad \mathcal{F}^{\square} := \{p \in \mathcal{C} : \mathcal{F} \,\square\, p\}\,.$$

It is easy to see that $\left({}^{\square}\mathcal{M}, ({}^{\square}\mathcal{M})^{\square}\right)$ is a weak factorisation system on the category of contexts. However, our axiom requires somewhat more than this, since the factorisation $f = pi$ we construct always has $p \in \mathcal{M}$ rather than merely $p \in ({}^{\square\square}\mathcal{M})$. The class $\mathcal{M}$ "fibrantly generates" our factorisation system.

Our factorisation axiom does not *directly* imply the existence of a weak factorisation system. In our case, we have classes $\mathcal{E}$ and $\mathcal{M}$ satisfying property (1), but not property (2): though we have $\mathcal{E} = {}^{\square}\mathcal{M}$, in general we only have $\mathcal{M} \subset \mathcal{E}^{\square}$. We can fix this by replacing $\mathcal{M}$ with the larger class $\mathcal{M}' = \mathcal{E}^{\square}$; then $(\mathcal{E}, \mathcal{M}')$ will form a weak factorisation system on the category of contexts $\mathcal{C}$. However, we cannot recover $\mathcal{M}$ from $\mathcal{M}'$, and since it is $\mathcal{M}$ that we are interested in rather than $\mathcal{M}'$, we will stick with the stronger formulation that we have given.

Remark. We have not so far shown how to reintroduce the "ambient context" for our new factorisation types. If we present the axioms type-theoretically, this is trivial; we merely add a Θ on the front of all our contexts and add definitional equalities saying that factorisation types are stable under substitution.

If we present them category-theoretically, on the other hand, we must be slightly more careful. Given a context Θ, we write $\mathcal{C}_\Theta$ for the full subcategory of the slice category $\mathcal{C}/\Theta$ on the objects $f\colon \Delta \to \Theta$ where f is a composite of maps from $\mathcal{M}$. In other words, $\mathcal{C}_\Theta$ is the category of "contexts in context Θ". Any map of contexts $F\colon \Theta' \to \Theta$ induces a substitution functor $F^*\colon \mathcal{C}_\Theta \to \mathcal{C}_{\Theta'}$, and there is an evident forgetful functor $U\colon \mathcal{C}_\Theta \to \mathcal{C}$. We write $\mathcal{M}_\Theta$ for the class of arrows of $\mathcal{C}_\Theta$ whose image under U lies in $\mathcal{M}$, and $\mathcal{E}_\Theta$ for ${}^{\square}(\mathcal{M}_\Theta)$. We can now state our factorisation axiom more precisely:

Axiom. *We can factorise any map $f \in \mathcal{C}_\Theta$ as $f = pi$ where $p \in \mathcal{M}_\Theta$ and $i \in \mathcal{E}_\Theta$; moreover, these factorisations are strictly preserved by the reindexing functors $F^*\colon \mathcal{C}_\Theta \to \mathcal{C}_{\Theta'}$.*

Despite the usefulness of our factorisation types, we cannot do everything we would like with them. As it stands, we cannot use them to capture the intensional product types; the reason for this is that intensional type theory is only *first order*, in the sense that we do not have judgements like

$$\big(x : A \vdash \phi(x) : B(x)\big) \vdash C(\phi) \ \mathsf{type}$$

available. In the next section, we shall describe a system with higher order types $(x : A)B$, in which the above judgement can be expressed, as

$$\phi : (x : A)B \vdash C(\phi) \ \mathsf{type}.$$

In this new system, we will be able to capture the intensional product type $\Pi(A, B)$ as the factorisation type of the context map

$$\big(\phi : (x : A)B\big) \to ().$$

2 A framework for type theory

The system we shall describe in this section can be seen as a superstructure which we erect around an intensional type theory; as well as judgements $\vdash A$ type we shall have judgements $\vdash A\ \mathsf{sort}$. The idea is that whilst the *types* continue to form a model of an intensional type theory, the *sorts* will form a model of a (extensional) dependently-typed lambda calculus. We view the types as an "intensional reflection" of the sorts; thus every type is a sort whilst every sort can be approximated by a type in a universal way.[1] All the constructions in the theory of types will arise by reflecting down the corresponding constructions in the theory of sorts.

This system is *not* the same as the "Logical Framework" which is commonly used to present intensional dependent type theory. This is also a (extensional) dependently-typed lambda calculus of *sorts* surrounding an intensional theory of *types*, but differs from our framework in that the theory of types is internal to the theory of sorts; we have $\mathsf{type} : \mathsf{sort}$ together with a "universal small map" $t : \mathsf{type} \vdash \mathsf{El}(t)\ \mathsf{sort}$, and all the structure *in* the theory of types is encoded as structure *on* type in the theory of sorts. So in this system, types are both *smaller* and *more intensional* than sorts.

In our system, by contrast, types are not smaller than sorts. The reflection of sorts into types will say (amongst other things) that every sort inductively generates a type, which immediately makes it clear that if we did have $\mathsf{type} : \mathsf{sort}$ then $\mathsf{type} : \mathsf{type}$ style paradoxes would await. A more correct intuition would be to think of the types as being the *inductively generated* sorts.

Since this framework is not standard, we shall give a thorough presentation of it from the ground up; we start by presenting the calculus of sorts. This can be summarised by saying that it is the framework for the monomorphic version of intensional type theory which is presented in [Nordström, et al., Part III], but without $\mathsf{type} : \mathsf{sort}$. We have four standard forms of judgement:

$$\Gamma \vdash A\ \mathsf{sort} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash A = B\ \mathsf{sort} \qquad \Gamma \vdash a = b : A$$

where Γ is a *well-formed context of sorts*, $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$. To say that Γ is well-formed, is to say that the following judgements hold:

$$\begin{aligned} &\vdash A_1\ \mathsf{sort} \\ x_1 : A_1 &\vdash A_2\ \mathsf{sort} \\ x_1 : A_1, x_2 : A_2 &\vdash A_3\ \mathsf{sort} \\ &\ldots \\ x_1 : A_1, \ldots, x_{n-1} : A_{n-1} &\vdash A_n\ \mathsf{sort}. \end{aligned}$$

We have other requirements for well-formed judgements:

- for the judgement $\Gamma \vdash a : A$ to be well-formed, we must have first the judgement $\Gamma \vdash A\ \mathsf{sort}$;
- for the judgement $\Gamma \vdash A = B\ \mathsf{sort}$ to be well-formed, we must have first the judgements $\Gamma \vdash A\ \mathsf{sort}$ and $\Gamma \vdash B\ \mathsf{sort}$;
- for the judgement $\Gamma \vdash a = b : A$ to be well-formed, we must have first the judgements $\Gamma \vdash a : A$ and $\Gamma \vdash b : A$.

We now give our rules of inference. For the sake of clarity, we omit premisses that can be inferred from the context: for instance, when we write the premise

[1] Though as we shall see, this universality does not amount to a reflection in the usual category-theoretic sense, since it is only universal "up to propositional equality".

$\Gamma \vdash a : A$, we implicitly presume also that $\Gamma \vdash A$ sort. We suppress any mention of a context that is common to both the premisses and the conclusion of a rule; finally, given a judgement $\vdash \mathcal{J}$ in an empty context, we omit the $\vdash$ entirely.

- *Assumption*

$$\frac{A \text{ sort}}{x : A \vdash x : A}$$

- *Equality of sorts*

$$\frac{A \text{ sort}}{A = A \text{ sort}} \qquad \frac{A = B \text{ sort}}{B = A \text{ sort}} \qquad \frac{A = B \text{ sort} \qquad B = C \text{ sort}}{A = C \text{ sort}}$$

- *Equality of elements*

$$\frac{a : A}{a = a : A} \qquad \frac{a = b : A}{b = a : A} \qquad \frac{a = b : A \qquad b = c : A}{a = c : A}$$

- *Sort rules*

$$\frac{a : A \qquad A = B \text{ sort}}{a : B} \qquad \frac{a = b : A \qquad A = B \text{ sort}}{a = b : B}$$

- *Substitution in sorts*

$$\frac{x : A, \Delta \vdash C \text{ sort} \qquad a : A}{\Delta[a/x] \vdash C[a/x] \text{ sort}} \qquad \frac{x : A, \Delta \vdash C \text{ sort} \qquad a = b : A}{\Delta[a/x] \vdash C[a/x] = C[b/x] \text{ sort}}$$

$$\frac{x : A, \Delta \vdash B = C \text{ sort} \qquad a : A}{\Delta[a/x] \vdash B[a/x] = C[a/x] \text{ sort}}$$

- *Substitution in elements*

$$\frac{x : A, \Delta \vdash c : C \qquad a : A}{\Delta[a/x] \vdash c[a/x] : C[a/x]} \qquad \frac{x : A, \Delta \vdash c : C \qquad a = b : A}{\Delta[a/x] \vdash c[a/x] = c[b/x] : C[a/x]}$$

$$\frac{x : A, \Delta \vdash b = c : C \qquad a : A}{\Delta[a/x] \vdash b[a/x] = c[a/x]}$$

This completes the list of core structural rules of the theory of sorts. We continue by adding in constructors for higher-order sorts.

- *Function sort formation*

$$\frac{A \text{ sort} \qquad x : A \vdash B \text{ sort}}{(x : A)B \text{ sort}} \qquad \frac{A_1 = A_2 \text{ sort} \qquad x : A_1 \vdash B_1 = B_2 \text{ sort}}{(x : A_1)B_1 = (x : A_2)B_2 \text{ sort}}$$

- *Abstraction*

$$\frac{x : A \vdash b : B}{(x)b : (x : A)B}$$

- *α-conversion*

$$\frac{x : A \vdash b : B \qquad a : A}{(x)b = (y)(b[y/x]) : (x : A)B} \quad y \text{ not free in } b$$

- *ξ-conversion*

$$\frac{x : A \vdash b_1 = b_2 : B}{(x)b_1 = (x)b_2 : (x : A)B}$$

- *Application*

$$\frac{f : (x : A)B \qquad a : A}{f \cdot a : B[a/x]} \qquad \frac{f_1 = f_2 : (x : A)B \qquad a_1 = a_2 : A}{f_1 \cdot a_1 = f_2 \cdot a_2 : B[a_1/x]}$$

- *β-conversion*

$$\frac{x : A \vdash b : B \qquad a : A}{(x)b \cdot a = b[a/x] : B[a/x]}$$

- *η-conversion*

$$\frac{f : (x : A)B}{f = (x)(f \cdot x) : (x : A)B}$$

We augment this extensional dependently typed lambda calculus of sorts with the following rules for types. We have one new form of judgement:

$$\Gamma \vdash A \ \mathsf{type}$$

where as before Γ is a context of sorts. We have that all types are sorts:

$$\frac{\Gamma \vdash A \ \mathsf{type}}{\Gamma \vdash A \ \mathsf{sort}.}$$

And that being a type is stable under substitution:

$$\frac{x : A, \Delta \vdash C \ \mathsf{type} \qquad a : A}{\Delta[a/x] \vdash C[a/x] \ \mathsf{type}}$$

Finally, we have laws for factorisation types. As before, we write $\mathbf{x} : \Gamma \vdash f(\mathbf{x}) : \Delta$ as shorthand for an arbitrary map of contexts $f \colon \Gamma \to \Delta$.

- *Factorisation type formation*:

$$\frac{\mathbf{x} : \Gamma \vdash f(\mathbf{x}) : \Delta}{\mathbf{y} : \Delta \vdash \Phi_f(\mathbf{y}) \ \mathsf{type}} \qquad \frac{\mathbf{x} : \Gamma \vdash f(\mathbf{x}) = g(\mathbf{x}) : \Delta}{\mathbf{y} : \Delta \vdash \Phi_f(\mathbf{y}) = \Phi_g(\mathbf{y}) \ \mathsf{sort}}$$

- *Factorisation type introduction*:

$$\frac{\mathbf{x} : \Gamma}{i(\mathbf{x}) : \Phi_f\big(f(\mathbf{x})\big)}$$

- *Factorisation type elimination*:

$$\frac{\mathbf{y} : \Delta,\ z : \Phi_f(\mathbf{y}) \vdash C(\mathbf{y}, z) \ \mathsf{type} \qquad \mathbf{x} : \Gamma \vdash d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big)}{\mathbf{y} : \Delta,\ z : \Phi_f(\mathbf{y}) \vdash J_d(\mathbf{y}, z) : C(\mathbf{y}, z)}$$

$$\frac{\mathbf{y} : \Delta,\ z : \Phi_f(\mathbf{y}) \vdash C(\mathbf{y}, z) \ \mathsf{type} \qquad \mathbf{x} : \Gamma \vdash d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big)}{\mathbf{y} : \Delta,\ z : \Phi_f(\mathbf{y}) \vdash J_d\big(f(\mathbf{x}), i(\mathbf{x})\big) = d(\mathbf{x}) : C\big(f(\mathbf{x}), i(\mathbf{x})\big).}$$

This concludes the list of formal axioms for our system, which we shall call system $\mathcal{S}$. What might not immediately obvious is that the types in this system form a full model of the (polymorphic) intensional dependent type theory $\mathcal{T}$. Let us show that this is the case. First observe that we can define the notion of a "context of types"; this is a context of sorts $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$ such that the following judgements hold:

$$\begin{aligned} &\vdash A_1 \ \mathsf{type} \\ x_1 : A_1 &\vdash A_2 \ \mathsf{type} \\ x_1 : A_1, x_2 : A_2 &\vdash A_3 \ \mathsf{type} \\ &\ldots \\ x_1 : A_1, \ldots, x_{n-1} : A_{n-1} &\vdash A_n \ \mathsf{type}. \end{aligned}$$

Using this notion, we can interpret the four basic judgements of $\mathcal{T}$:

- To say that A is a type in context Γ is to say that Γ is a context of types and that $\Gamma \vdash A$ type holds in $\mathcal{S}$;
- To say that A and B are equal types in context Γ is to say that Γ is a context of types and that $\Gamma \vdash A = B$ sort holds in $\mathcal{S}$;
- To say that a is an element of the type A in context Γ is to say that Γ is a context of types and that $\Gamma \vdash a : A$ holds in $\mathcal{S}$;
- To say that a and b are equal elements of the type A in context Γ is to say that Γ is a context of types and that $\Gamma \vdash a = b : A$ holds in $\mathcal{S}$.

So we have a subsystem of types and contexts of types which is trivially checked to satisfy all the core structural rules of $\mathcal{T}$. What about the type constructors of $\mathcal{T}$? All of these will arise as a result of the factorisation types.

- The unit type 1 arises as the factorisation type of $! \colon () \to ()$;
- Given a type A, the identity type $x, y : A \vdash \mathsf{Id}_A(x, y)$ type arises as the factorisation type of $\Delta \colon (x : A) \to (x : A,\, y : A)$;
- Given a judgement $x : A \vdash B$ type, the intensional sum type $\Sigma(A, B)$ arises as the factorisation type of $! \colon \big(a : A, b : B(a)\big) \to ()$;
- Given a judgement $x : A \vdash B$ type, the intensional product type $\Pi(A, B)$ arises as the factorisation type of $! \colon \big(\phi : (x : A)B\big) \to ()$.

3 Categorical models of system $\mathcal{S}$

Definition 5 (Taylor)**.** In a category $\mathcal{C}$, a **class of display maps** $\mathcal{D}$ is a subclass of the arrows of $\mathcal{C}$ such that pullbacks of $\mathcal{D}$-maps along $\mathcal{C}$-maps exist, and are $\mathcal{D}$-maps.

Definition 6. A categorical model for system $\mathcal{S}$ is given by the following data:

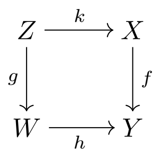
- A category $\mathcal{C}$ (modelling "contexts of sorts") with finite products;
- A class of display maps $\mathcal{D}$ in $\mathcal{C}$ (modelling "dependent projections of sorts"), and
- A class of display maps $\mathcal{M} \subset \mathcal{D}$ (modelling "dependent projections of types").

Given an object $X \in \mathcal{C}$, we single out three full subcategories of the slice category $\mathcal{C}/X$:

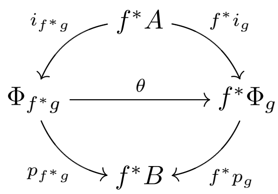
- $\mathcal{T}_X$ (modelling "types in context X") is the full subcategory whose objects are $\mathcal{M}$-maps;
- $\mathcal{S}_X$ (modelling "sorts in context X") is the full subcategory whose objects are $\mathcal{D}$-maps, and
- $\mathcal{C}_X$ (modelling "contexts in context X") is the full subcategory whose objects are composites of zero or more $\mathcal{D}$-maps.

We also write $\mathcal{M}_X$ for the subclass of arrows of $\mathcal{C}_X$ whose image under the forgetful functor $\mathcal{C}_X \to \mathcal{C}$ lies in $\mathcal{M}$, and write $\mathcal{E}_X$ for ${}^{\square}(\mathcal{M}_X)$. We now require that:

- For any map $f\colon X \to Y$ in $\mathcal{D}$, the pullback functor $f^*\colon \mathcal{S}_Y \to \mathcal{S}_X$ has a right adjoint Π_f, and these right adjoints satisfy the *Beck-Chevalley* condition: given a pullback square

  with $f, g \in \mathcal{D}$, the canonical natural transformation $h^*\Pi_f \Rightarrow \Pi_g k^*$ is a natural isomorphism.
- Every map $g\colon A \to B$ in $\mathcal{C}_X$ has a chosen factorisation $f\colon A \xrightarrow{i} \Phi_g \xrightarrow{p} B$ where $p \in \mathcal{M}_X$ and $i \in \mathcal{E}_X$; moreover, given a context map $f\colon Y \to X$ in $\mathcal{C}$, the pullback functor $f^*\colon \mathcal{C}_X \to \mathcal{C}_Y$ preserves the factorisations, in that there is an isomorphism $\theta\colon \Phi_{f^*g} \to f^*\Phi_g$ making the following diagram commute in $\mathcal{C}_Y$:

Remark. To be faithful with the type theory, we should require these factorisations to be *strictly* preserved. However, in keeping with the rest of the definition, which is fairly laissez-faire about coherence, we only demand isomorphisms. If we wanted to be really precise we would restate everything in terms of split fibrations *à la* Jacobs. However the translation is routine and the display map formulation is more convenient so we shall stick to it.

Note that there is actually a further weakening that we could make, namely to require that f^* sends the *chosen* factorisation in $\mathcal{M}_X$ to *some* factorisation in $\mathcal{M}_Y$; that is, we demand merely that $f^*p_g \in \mathcal{M}_Y$[2] and $f^*i_g \in \mathcal{E}_Y$.

Remark. We said earlier that the types would be an "intensional reflection" of the kinds. We make this precise as follows. Fix an object $X \in \mathcal{C}$; then

[2]which is automatic.

since $\mathcal{M} \subset \mathcal{D}$ ("types are sorts") we have an inclusion functor $\mathcal{T}_X \to \mathcal{S}_X$. This inclusion functor is trying very hard to have a left adjoint, which sends a $\mathcal{D}$-map $(f\colon A \to X)$ to the $\mathcal{M}$-map $(p_f\colon \Phi_f \to X)$.

Unfortunately, this operation is not functorial on the nose, but only "up to propositional equality"; and even if it were, then it is not a left adjoint on the nose, because one of the triangle identities does not commute on the nose, but again only "up to propositional equality".