The groupoid model of type theory

Ethan Lewis, Max Bohnet

1 The Logical Framework

Before we can discuss the groupoid model, it is important that we take a cursory glance at the logical framework in which our type theory is formalized. In this framework, there are three ways to form types:

$$\Gamma \vdash Set : Type \qquad \frac{\Gamma \vdash A : Set}{\Gamma \vdash El(A) : Type} \qquad \frac{\Gamma \vdash A : Type \qquad \Gamma, a : A \vdash B : Type}{(a : A)B : Type}$$

We will often write $a : A$ as an abbreviation for $a : El(A)$, and $A \to B$ as an abbreviation for $(a : A)B$ if a does not occur in B. One can think of Set as a type of types.

Instead of giving formation, introduction, elimination, and computation rules, we define sets by giving the types of their associated operations. For example, the types of the identity set operations are as follows:

$$\begin{aligned}
\mathsf{Id} &: (A : Set)A \to A \to Set \\
\mathsf{refl} &: (A : Set)(a : A)\mathsf{Id}(A, a, a) \\
J &: (A : Set)(C : (a_1, a_2 : A)\mathsf{Id}(A, a_1, a_2) \to Set)\big((a : A)C(a, a, \mathsf{refl}(A, a))\big) \\
&\qquad \to (a_1, a_2 : A)\big(s : \mathsf{Id}(A, a_1, a_2)\big)C(a_1, a_2, s) \\
J\big(A, C, d, a, a, \mathsf{refl}(A, a)\big) &= d(a) : C\big(a, a, \mathsf{refl}(A, a)\big)
\end{aligned}$$

Note that there is a natural correspondence between the types of these terms and the typical formation, introduction, elimination, and computation rules for the identity type.

2 The Groupoid Model

The primary purpose of the groupoid model is to show that identity proofs need not be unique. Formally speaking, its purpose is to show that the type UIP (defined below) is empty.

Definition 1. Uniqueness of identity proofs (UIP) is the claim that for every type A and every $a_1, a_2 : A$, if $s_1, s_2 : a_1 = a_2$, then $s_1 = s_2$. In other words, UIP is the following type:

$$(A : Set)(a_1, a_2 : A)\big(s_1, s_2 : \mathsf{Id}(a_1, a_2)\big)\mathsf{Id}\big(\mathsf{Id}(A, a_1, a_2), s_1, s_2\big)$$

With this in mind, we turn our attention to the groupoid model. The model is a category with families. Recall the definition of a category with families:

Definition 2. A category with families consists of the following data:

1. A category $\mathcal{C}$ of contexts and substitutions with terminal object $[]$ corresponding to the empty context.

2. A collection-valued functor $Ty : \mathcal{C}^{op} \to \mathcal{S}et$ associating with each context Γ the collection of types depending on it. If $f : \Delta \to \Gamma$ and $A \in Ty(\Gamma)$, one writes $A\{f\}$ for $Ty(f)(A)$. The type $A\{f\}$ corresponds to the substitution of f into A.

3. For every $\Gamma \in \mathcal{C}$ and $A \in Ty(\Gamma)$, a collection of terms $Tm(\Gamma, A)$, together with a substitution function $Tm(f, A) : Tm(\Gamma, A) \to Tm(\Delta, A\{f\})$ functorial in $f : \Delta \to \Gamma$ in the obvious sense.

4. For every $A \in Ty(\Gamma)$, a context extension $\Gamma.A$, which has the property that the hom-set $\mathcal{C}(\Delta, \Gamma.A)$ and $\{(f, M) \mid f : \Delta \to \Gamma \text{ and } M \in Tm(\Delta, A\{f\})\}$ are isomorphic naturally in Δ.

5. Operations corresponding to the desired type, set, and term formers.

Our particular model is as follows:

Definition 3. The groupoid model consists of the following data:

1. The (large) category GPD of groupoids. (A groupoid is a category where every morphism is an isomorphism. We will also be interested in the category Gpd whose objects are the small groupoids, and whose arrows are the isomorphisms between these small groupoids.)

2. The functor $Ty : GPD^{op} \to \mathcal{S}et$, where $Ty(\Gamma)$ is the collection of families of groupoids over Γ, and $Ty(f)(A) = A \circ f$. (A family of groupoids over Γ is a functor $A : \Gamma \to GPD$. We will also be interested in the functor $Se : Gpd^{op} \to \mathcal{S}et$, where $Se(\Gamma)$ is the collection of small families of groupoids over Γ [i.e. functors $A : \Gamma \to Gpd$], and $Se(f)(A) = A \circ f$.)

3. For every $\Gamma \in GPD$ and $A \in Ty(\Gamma)$, a collection $Tm(\Gamma, A)$ of dependent objects of A together with the substitution function $Tm(f, A) : Tm(\Gamma, A) \to Tm(\Delta, A\{f\})$ given by $Tm(f, A)(a) = a \circ f$. (A dependent object of $A \in Ty(\Gamma)$ consists of the following data:

   a) An $A(\gamma)$-object $M(\gamma)$ for each $\gamma \in \Gamma$.

   b) For every morphism $p : \gamma \to \gamma'$, an $A(\gamma')$-morphism $M(p) : p \cdot M(\gamma) \to M(\gamma')$ [recall that $p \cdot \_ : A(\gamma) \to A(\gamma')$ is the functor given by $A(p)$] such that $M(id_\gamma) = id_{M(\gamma)}$ and $M(p' \circ p) = M(p') \circ (p' \cdot M(p))$.)

4. For every $A \in Ty(\Gamma)$, the context extension $\Gamma.A$, where the objects of $\Gamma.A$ are pairs (γ, a) such that $\gamma \in \Gamma$ and $a \in A(\gamma)$, and a morphism $(\gamma, a) \to (\gamma', a')$ is a pair (p, q), where $p \in \Gamma(\gamma, \gamma')$ and $q \in A(\gamma')(p \cdot a, a')$. The identity on (γ, a) is (id_γ, id_a), composition is given by $(p', q') \circ (p, q) = (p' \circ p, q' \circ (p' \cdot q))$, and the inverse of (p, q) is $(p^{-1}, p^{-1} \cdot q^{-1})$.

5. Operations corresponding to the desired type, set, and term formers. (We will only discuss some of these in detail.)

3 Interpreting Identity Sets

To better understand the groupoid model and why it is a countermodel of UIP, we will give interpretations of the identity set operations. Since identity sets will be interpreted as discrete groupoids, we must first give the definition of a discrete groupoid:

Definition 4. For a set X, the discrete groupoid $\triangle(X)$ has as objects the elements of x and its only morphisms are the identities. Instead of writing id_x for the identity on x, we will write $\star$. Note that $\star : x \to y$ if and only if $x = y$.

Now we can provide the desired interpretations:

3.1 Id

By currying (this will be explained in more detail later on), it suffices to define a small family Id over the groupoid $[A : Set, a_1, a_2 : A]$. The objects of this groupoid are triples (A, a_1, a_2), where A is a small groupoid and $a_1, a_2 \in A$. A morphism $(A, a_1, a_2) \to (A', a'_1, a'_2)$ is a triple (p, q_1, q_2), where $p \in Gpd(A, A')$ and $q_i \in A'(p(a_i), a'_i)$. We define the family Id as follows:

$$\mathsf{Id}(A, a_1, a_2) = \triangle\big(A(a_1, a_2)\big)$$
$$\mathsf{Id}(p, q_1, q_2)(s) = q_2 \circ p(s) \circ q_1^{-1}$$

One can easily verify that Id is a small family.

3.2 Refl

Let Id_{diag} be given by

$$\mathsf{Id}_{diag}(A, a) = \mathsf{Id}(A, a, a)$$
$$\mathsf{Id}_{diag}(p, q) = \mathsf{Id}(p, q, q)$$

By currying, it suffices to define a dependent object refl of Id_{diag} over the groupoid $[A : Set, a : A]$. The objects of the groupoid $[A : Set, a : A]$ are pairs (A, a), where A is a small groupoid and $a \in A$. A morphism $(A, a) \to (A', a')$ is a pair (p, q), where $p \in Gpd(A, A')$ and $q \in A'(p(a), a')$. To define refl, let the object part be given by $\mathsf{refl}(A, a) = id_a$. For the morphism part, recall that the morphisms in a discrete groupoid are $\star : x \to x$ for every x in the groupoid and that $\star : x \to$

y if and only if $x = y$. Since $\mathsf{Id}_{diag}(A', a')$ is the discrete groupoid $\triangle(A(a', a'))$, it must be that $\mathsf{refl}(p, q) = \star : (p, q) \cdot \mathsf{refl}(A, a) \to \mathsf{refl}(A', a')$. Such a morphism exists if $(p, q) \cdot \mathsf{refl}(A, a) = \mathsf{refl}(A', a')$, which is indeed the case:

$$\begin{aligned}(p, q) \cdot \mathsf{refl}(A, a) &= q \circ p\big(\mathsf{refl}(A, a)\big) \circ q^{-1} \\ &= q \circ p(id_a) \circ q^{-1} \\ &= q \circ id_{p(a)} \circ q^{-1} \\ &= q \circ q^{-1} \\ &= id_{a'} \\ &= \mathsf{refl}(A', a')\end{aligned}$$

One can easily verify the remaining morphism requirements on refl.

3.3 J

By currying, it suffices to define a dependent object J of C over the groupoid $[\Gamma, a_1, a_2 : A, s : \mathsf{Id}(A, a_1, a_2)]$, where

$$\Gamma = \big[A : Set, C : \big(a_1, a_2 : A, s : \mathsf{Id}(A, a_1, a_2)\big)Set, d : (a : A)C\big(a, a, \mathsf{refl}(A, a)\big)\big]$$

To be explicit, the groupoid $[\Gamma, a_1, a_2 : A, s : \mathsf{Id}(A, a_1, a_2)]$ has as objects tuples (γ, a_1, a_2, s), where γ is a small groupoid, $a_1, a_2 \in A(\gamma)$, and $s \in Id(A(\gamma), a_1, a_2)$. A morphism $(\gamma, a_1, a_2, s) \to (\gamma', a_1', a_2', s')$ is a tuple $(p, q_1, q_2, \star)$, where $p \in Gpd(\gamma, \gamma')$, $q_i \in A(\gamma')(p \cdot a_i, a_i')$, and $\star \in \mathsf{Id}(A(\gamma'), a_1', a_2')\big((p, q_1, q_2) \cdot s, s'\big)$, and composition of morphisms is given by

$$(p', q_1', q_2', \star) \circ (p, q_1, q_2, \star) = (p' \circ p, q_1' \circ (p' \cdot q_1), q_2' \circ (p' \cdot q_2), \star)$$

Note that we can obtain C and a small family $A \in Se(\Gamma)$ through projection on $[\Gamma, a_1, a_2 : A, s : \mathsf{Id}(A, a_1, a_2)]$. Similarly, we can also obtain a dependent object d of C_{diag} over the groupoid $[\Gamma, a : A]$, where

$$\begin{aligned}C_{diag}(\gamma, a) &= C\big(\gamma, a, a, \mathsf{refl}(A(\gamma), a)\big) \\ C_{diag}(p, q) &= C(p, q, q, \star)\end{aligned}$$

To define the object part of J, let $u = (\gamma, a_1, a_2, s)$ be an object of $[\Gamma, a_1, a_2 : A, s : \mathsf{Id}(A, a_1, a_2)]$, and define $f(u) = (id_\gamma, id_{a_1}, s, \star)$, where $\star : s \to s$. Note that $f(u) : \big(\gamma, a_1, a_1, \mathsf{refl}(A(\gamma), a_1)\big) \to (\gamma, a_1, a_2, s)$ because

$$\begin{aligned}(id_\gamma, id_{a_1}, s) \cdot \mathsf{refl}(A(\gamma), a_1) &= s \circ A(id_\gamma)\big(\mathsf{refl}(A(\gamma), a_1)\big) \circ id_{a_1}^{-1} \\ &= s \circ id_{A(\gamma)}(id_{a_1}) \circ id_{a_1} \\ &= s\end{aligned}$$

Therefore, since $d(\gamma, a_1) \in C(\gamma, a_1, a_1, \mathsf{refl}(A(\gamma), a_1))$, we can define $J(u) = f(u) \cdot d(\gamma, a_1) \in C(\gamma, a_1, a_2, s)$.

For the morphism part, let $u = (\gamma, a_1, a_2, s)$ and $u' = (\gamma', a_1', a_2', s')$ be objects of $[\Gamma, a_1, a_2 : A, \mathsf{Id}(A, a_1, a_2)]$, and let $h = (p, q_1, q_2, \star) : u \to u'$. Define $J(h) = f(u') \cdot d(p, q_1)$. I claim that $J(h) : h \cdot J(u) \to J(u')$ in $C(u')$. To see why, observe that $(p, q_1) : (\gamma, a_1) \to (\gamma', a_1')$. Since d is a dependent object of C_{diag}, we have that $d(p, q_1) : (p, q_1) \cdot d(\gamma, a_1) \to d(\gamma', a_1')$, so it follows from the definition of C_{diag} that $d(p, q_1) : (p, q_1, q_1, \star) \cdot d(\gamma, a_1) \to d(\gamma', a_1')$. Thus,

$$\begin{aligned} f(u') \cdot d(p, q_1) &: f(u') \cdot \big((p, q_1, q_1, \star) \cdot d(\gamma, a_1)\big) \to f(u') \cdot d(\gamma', a_1') \\ J(h) &: \big(f(u') \circ (p, q_1, q_1, \star)\big) \cdot d(\gamma, a_1) \to J(u') \end{aligned}$$

Now we want to show that $f(u') \circ (p, q_1, q_1, \star) = h \circ f(u)$ because then

$$\begin{aligned} \big(f(u') \circ (p, q_1, q_1, \star)\big) \cdot d(\gamma, a_1) &= \big(h \circ f(u)\big) \cdot d(\gamma, a_1) \\ &= h \cdot \big(f(u) \cdot d(\gamma, a_1)\big) \\ &= h \cdot J(u) \end{aligned}$$

which implies that $J(h) : h \cdot J(u) \to J(u')$. But before showing $f(u') \circ (p, q_1, q_1, \star) = h \circ f(u)$, recall that $(p, q_1, q_2, \star) : u \to u'$, so $\star : q_2 \circ (p \cdot s) \circ q_1^{-1} \to s'$. Therefore, $q_2 \circ (p \cdot s) \circ q_1^{-1} = s'$, so $q_2 \circ (p \cdot s) = s' \circ q_1$. Using this fact, we can now show that $f(u') \circ (p, q_1, q_1, \star) = h \circ f(u)$:

$$\begin{aligned} f(u') \circ (p, q_1, q_1, \star) &= (id_{\gamma'}, id_{a_1'}, s', \star) \circ (p, q_1, q_1, \star) \\ &= (p, q_1, s' \circ q_1, \star) \\ &= (p, q_1, q_2 \circ (p \cdot s), \star) \\ &= (p, q_1, q_2, \star) \circ (id_\gamma, id_{a_1}, s, \star) \\ &= h \circ f(u) \end{aligned}$$

Thus, $J(h) : h \cdot J(u) \to J(u')$. Demonstrating that J satisfies the remaining morphism requirements is tedious but straightforward.

3.4 J = d

It follows from our definition of J that the desired equality between J and d holds:

$$\begin{aligned} J\big(\gamma, a, a, \mathsf{refl}(A(\gamma), a)\big) &= f\big(\gamma, a, a, \mathsf{refl}(A(\gamma), a)\big) \cdot d(\gamma, a) \\ &= (id_\gamma, id_a, id_a, \star) \cdot d(\gamma, a) \\ &= d(\gamma, a) \\ J(p, q, q, \star) &= f\big(\gamma', a', a', \mathsf{refl}(A(\gamma'), a')\big) \cdot d(p, q) \\ &= (id_{\gamma'}, id_{a'}, id_{a'}, \star) \cdot d(p, q) \\ &= d(p, q) \end{aligned}$$

4 UIP Revisited

Now that we have the interpretation of identity sets, it becomes clear why the groupoid model is a countermodel of UIP:

Theorem 1. *The type UIP is empty.*

Proof. Assume to the contrary that UIP is nonempty. Then there exists $u \in Tm(\text{UIP})$. Let A be the group $\mathbb{Z}_2$ viewed as a one-object groupoid. More specifically, A has one object $\star$ and distinct morphisms $id_\star$ and p such that $p \circ p = id_\star$. Since $u \in Tm(\text{UIP})$, it follows that $u(A, \star, \star, p, id_\star) \in \mathsf{Id}\big(\mathsf{Id}(A, \star, \star), p, id_\star\big)$, but $\mathsf{Id}\big(\mathsf{Id}(A, \star, \star), p, id_\star\big)$ is empty because $p \neq id_\star$. Hence, UIP is empty. $\dashv$

5 $\prod$-types

For this section, let $A \in Ty(\Gamma), B \in Ty(\Gamma.A)$. First we define groupoid structure on $Tm(A)$. The leading idea here is that terms of a type over Γ correspond $1-1$ to sections of the canonical projection $\mathsf{p}_A : \Gamma.A \to \Gamma$ — which in the groupoid model are functors – so we can define a groupoid of terms in terms of a category of functors and natural isomorphisms.

Definition 5 (Groupoid of terms)**.** The category $Tm(A)$ is then defined as follows:

Objects $Tm(A)$

Morphisms Let $M \in Tm(A)$. Define a functor $\overline{M} : \Gamma \to \Gamma.A$ by letting

$$\overline{M}(\gamma) = (\gamma, M(\gamma)) \qquad\qquad \overline{M}(p) = (p, M(p))$$

A morphism $\tau : M \to N$ is a family of morphism $\tau_\gamma : M(\gamma) \to N(\gamma)$ s.t. $\overline{\tau} := \{(id_\gamma, \tau_\gamma)\}_{\gamma\in\Gamma}$ is a natural transformation $\overline{\tau} : \overline{M} \Rightarrow \overline{N}$.

Since $M(\gamma), N(\gamma) \in A(\gamma) \in GPD$ we have τ_γ^{-1} for every τ_γ. It is easy to check that $\tau^{-1} := \{\tau_\gamma^{-1}\}_{\gamma\in\Gamma}$ is a morphism and an inverse to τ. Hence $Tm(A)$ is a groupoid.

Definition 6 (Π_{LF}, λ)**.** Let $\gamma \in \Gamma$. Define $\hat{\gamma} : A(\gamma) \to \Gamma.A$ by $\hat{\gamma}(a) = (\gamma, a), \hat{\gamma}(p : a \to a') = (id_\gamma, p)$.
Let $B_\gamma : B\{\hat{\gamma}\}$. Then the functor $\Pi_{LF}(A, B) : \Gamma \to GPD$ is defined as follows:

$$\Pi_{LF}(A,B)(\gamma) = Tm(B_\gamma)$$
$$\Pi_{LF}(A,B)(p : \gamma \to \gamma') =: p \cdot \_ : Tm(B_\gamma) \to Tm(B'_\gamma)$$

$p \cdot \_$ on objects ($a, a' \in A(\gamma')$):

$$(p \cdot M)(a) = (p, id_a) \cdot_B M(p^{-1} \cdot_A a)$$
$$(p \cdot M)(q : a \to a') = (p, id_a) \cdot_B M(p^{-1} \cdot_A q)$$

$p \cdot \_$ on morphisms $\tau : M \to M'$

$$(p \cdot \tau) : (p \cdot M) \to (p \cdot M')$$
$$(p \cdot \tau)_a = (p, id_a) \cdot_B \tau_{p^{-1} \cdot_A a}$$

Now let $M \in Tm(B)$. We define a term $\lambda_{A,B}(M) \in Tm(\Pi_{LF}(A, B))$. First we need an object $\lambda_{A,B}(M)(\gamma) \in \Pi_{LF}(A, B)(\gamma) = Tm(B_\gamma)$, i.e. another term, given by:

$$\lambda_{A,B}(M)(\gamma)(a) = M(\gamma, a) \qquad \lambda_{A,B}(M)(\gamma)(q) = M(id_\gamma, q)$$

Now we need for $p : \gamma \to \gamma'$ a morphism $p \cdot_{\Pi_{LF}(A,B)} \lambda_{A,B}(M)(\gamma) \to \lambda_{A,B}(M)(\gamma')$, i.e. a family indexed by $A(\gamma')$ inducing a natural transformation between the corresponding functors. The component at $a \in A(\gamma')$ is given by

$$(\lambda_{A,B}(M)(p))_a := M(p, id_a)$$

This gives us the interpretation for λ-abstraction.

Conversely, given a term $N \in Tm(\Pi(A, B))$ we want to recover its 'matrix', i.e. the dependent term $\lambda^{-1}_{A,B}(N) \in Tm(B)$. We define this on objects by

$$\lambda^{-1}_{A,B}(N)(\gamma, a) = N(\gamma)(a)$$

Let $(p, q) : (\gamma, a) \to (\gamma', a')$ be a morphism (in $\Gamma.A$), i.e. $p : \gamma \to \gamma'$ an $q : p \cdot a \to a'$. We need to define $\lambda^{-1}_{A,B}(N)(p, q) : (p, q) \cdot \lambda^{-1}_{A,B}(N)(\gamma, a) \to \lambda^{-1}_{A,B}(N)(\gamma', a')$, i.e. a morphism $(p, q) \cdot N(\gamma)(a) \to N(\gamma')(a')$. This is given by

$$\lambda^{-1}_{A,B}(N)(p, q) := N(\gamma')(q) \circ (id_{\gamma'}, q) \cdot N(p)_{p \cdot a}$$

Application of N to $O \in Tm(A)$ can now be defined as $\lambda^{-1}_{A,B}(N)\{\overline{O}\}$.

Fact 1. *The groupoid model supports* Π*-types.*

6 The universe of sets

Definition 7 (Universe of sets and its elements)**.** Set : $[] \to GPD, * \mapsto Gpd$, i.e. the groupoid of all metatheoretically small groupoids with isomorphisms as morphisms. Since they are isomorphic, we can identify $[].Set$ and Gpd.
$El : Gpd \hookrightarrow GPD$. If A is a small family, i.e. $A : \Gamma \to Gpd$, then $A = [\![[]\!]El(A)] = El\{A\} = El \circ A : \Gamma \to GPD$, so we can treat small families over Γ as a subset $Se(\Gamma) \subsetneq Ty(\Gamma)$ of all types over Γ.

Since the groupoid $\Pi_{LF}(A, B)$ is small if A and B are, we can interpret Π-sets simply as small Π-types.

7 Functional extensionality

Theorem 2. *The type*

$$Fun\_Ext \overset{def}{=} (A : Set)(B : (a : A)Set)(f, g : \Pi(A, B))((a : A)Id(B(a), f(a), g(a))) \to Id(\Pi(A, B), f, g)$$

is inhabited in the groupoid model

Proof. Let $\Gamma := [A : Set, B : (a : A)Set, f, g : \Pi(A, B)]$ and $PE := \Pi_{LF}(A, \mathsf{Id}(B(a), f(a), g(a)))$. Let $\gamma \in \Gamma$. The groupoid $\mathsf{Id}(\Pi_{LF}(A, B), f, g)(\gamma)$ is defined as the discrete groupoid over the set $hom_{\Pi_{LF}(A,B)(\gamma)}(f(\gamma), g(\gamma))$. More precisely, $f(\gamma), g(\gamma) \in Tm(B_\gamma)$, so morphisms between them are natural transformations indexed by $A(\gamma)$. Objects of $PE(\gamma)$ are functors $M : A(\gamma) \to \triangle(hom(f(\gamma), g(\gamma))$, i.e. $M(a)$ is a morphism $f(\gamma)(a) \to g(\gamma)(a)$ and $M(q) : (id_\gamma, q) \cdot M(a) \to M(a')$ since the identity set is a discrete groupoid, this yields the same commuting diagram, with $M(a)$ in place of τ_a. This induces an isomorphism between $PE(\gamma)$ and $\mathsf{Id}(\Pi_{LF}(A, B))(\gamma)$, which is moreover natural in γ, thus establishing a natural isomorphism between the functors PE and $\mathsf{Id}(\Pi_{LF}(A, B))$. ⊣

In fact, this isomorphism can be uniquely (up to propositional equality) specified by adding three axioms to the theory, which are true in the groupoid model.

8 Universe Extensionality

It is possible to add more universes than that of all types (*Set*) to the theory, together with type formers for all the types it should be closed under, in particular identity sets. This allows us to express identity of types. If we have a subuniverse V of the universe we used to interpret *Set* available in our metatheory, we can interpret this additional universe by $Gpd(V)$ or $Gpd_\triangle(V)$, defined analogously with Gpd under the additional proviso that the sets of objects and morphisms of the groupoids be in V. We then have the following

Fact 2. *If U is interpreted by $Gpd_\triangle(V)$ then identity types $\mathsf{Id}(U, A, B)$ are isomorphic to the isomorphism type*

$$Iso(A, B) \overset{def}{=} \Sigma([f : A \to B]\Sigma([g : B \to A]\mathsf{Id}(g \circ f, id) \times \mathsf{Id}(f \circ g, id)$$

This represents a restricted form of univalence.

9 Exercise

The point of this exercise is to give a partial interpretation of sum types in the groupoid model.

1. We first want to define an interpretation for

   $$+ : Set \to Set \to Set$$

   By currying it suffices to find a family over $Gpd \times Gpd$, i.e. a functor $+ : Gpd \times Gpd \to GPD$.

   a) Recall that the objects of $Gpd \times Gpd$ are pairs (A, B) with $A, B \in Gpd$ and a morphism $(A, B) \to (A', B')$ is a pair of morphisms $(f : A \to A', g : B \to B')$, with obvious identities composition defined pointwise.

      Check that $Gpd \times Gpd$ so defined is indeed a groupoid.

   b) Recall that the objects of the coproduct of A and B are $\{(0, a) \mid a \in Ob(A)\} \cup \{(1, b) \mid b \in Ob(B)\}$ and a morphism $(0, a) \to (0, a')$ is a morphism $a \to a'$ in A, and a morphism $(1, b) \to (1, b')$ is a morphism $b \to b'$ in B with the obvious identities and compositions.

      The functor $+$ on objects will be given by $(A, B) \mapsto A + B$, where $A + B$ is the coproduct of A and B.

      Give the action of $+$ on morphisms and check functoriality.

2. Next we want to define an interpretation for the left inclusion map

   $$i : (A, B : Set)A \to A + B$$

   By currying it suffices to define a dependent object i over the family $+\{p_A\} : [A : Set, B : Set, a : A] \to GPD$.

   The groupoid $[A : Set, B : Set, a : A]$ has as objects triples (X, Y, x) where $X, Y \in Gpd$ and $x \in X$. A morphism $(X, Y, x) \to (X', Y', x')$ is a triple (f, g, h) where $f : X \to X'$ and $g : Y \to Y'$ are isomorphisms of groupoids, and $h : f(x) \to x'$.

   The family $+\{p_A\} : [A : Set, B : Set, a : A] \to GPD$ is the following functor:

   $$+\{p_A\}(X, Y, x) = X + Y \qquad\qquad +\{p_A\}(f, g, h) = +(f, g)$$

   Define a suitable dependent object of $+\{p_A\}$, checking all the conditions.

3. (Optional!) Define suitable interpretations for

$$j : (A, B : Set)B \to A + B$$
$$D : (A, B : Set)(C : (A + B) \to Set)((a : A)C(i(a))) \to ((b : A)C(j(b))) \to (c : A + B)C(c)$$

such that

$$D(d, e, i(a)) = d(a) : C(i(a))$$
$$D(d, e, j(b)) = d(b) : C(j(b))$$