Logical Methods in Computer Science
Volume 21, Issue 3, 2025, pp. 32:1–32:84
https://lmcs.episciences.org/

Submitted Jul. 10, 2023
Published Sep. 26, 2025

RELATING HOMOTOPY EQUIVALENCES TO CONSERVATIVITY IN DEPENDENT TYPE THEORIES WITH COMPUTATION AXIOMS

MATTEO SPADETTO

University of Nantes, France
e-mail address: matteo.spadetto.42@gmail.com

Abstract. We prove a conservativity result for extensional type theories over propositional ones, i.e. dependent type theories with propositional computation rules, or computation axioms, using insights from homotopy type theory. The argument exploits a notion of canonical homotopy equivalence between contexts, and uses the notion of a category with attributes to phrase the semantics of theories of dependent types. Informally, our main result asserts that, for judgements essentially concerning h-sets, reasoning with extensional or propositional type theories is equivalent.

1. Introduction

In recent decades, dependent type theory has emerged as a powerful tool in the foundations of mathematics. Dependent types, i.e. types varying over the terms of other types, allow for precise and expressive specifications of mathematical statements and proofs, and provide a formal language for reasoning about them [NPS01, CH88]. This is not just of theoretical significance, as dependent types also have practical applications in fields such as software verification, where proof assistants like Coq and Agda have been successfully employed. More recently, the field has seen significant developments, based on the numerous insights provided by the foundational work of Vladimir Voevodsky in univalent foundations. This includes the emergence of Homotopy Type Theory [Uni13] and the Univalent Foundations program [Voe15], which provide a new understanding of the concept of equality. The focus of this paper is on the notion of conservativity between two different theories of dependent types, exploiting the insights coming from Homotopy Type Theory.

Suppose that we are given two dependent type theories T_1 and T_2 such that the inference rules, hence the type constructors, of T_1 are contained (or inferable) in T_2. In other words,

Key words and phrases: conservativity, homotopy equivalence, dependent type theory, propositional computation rule, category with attributes.

Earlier preprint versions of this article were titled *A conservativity result for homotopy elementary types in dependent type theory* and *Relating homotopy equivalences to conservativity in dependent type theories with propositional computation*.

Research supported by a School of Mathematics full-time EPSRC Doctoral Training Partnership Studentship 2019/2020, by an E-COST-MEETING-CA20111 WG6 2022, and by an E-COST-CA20111 Short-Term Scientific Mission Grant.

DOI:10.46298/LMCS-21(3:32)2025

let us assume that T_2 extends T_1. The first question that one can ask is whether T_1 and T_2 can deduce the same judgements, in which case the theories actually coincide (or are logically equivalent, respectively). However, even when this is not the case, one can still ask a question which is perhaps even more interesting: whether the two theories prove the same *statements*. By following the general interpretation of *statements as types* and *proofs as terms* (see [How80, ML84, Uni13] for more details), this property corresponds to the following: the theory T_1 considers any given type inhabited, provided it is inhabited according to T_2, i.e. whenever T_2 proves a term judgement $t : T$, then T_1 proves a term judgement of the form $\tilde{t} : T$. When this happens, we say that T_2 is *conservative* over T_1, meaning that *by "reinforcing" the theory* T_1 *with the rules of* T_2 *one does not risk to modify—namely increase—the deductive power of* T_1.

The property of conservativity was studied by Martin Hofmann [Hof95a, Hof96] for an intensional type theory with *extensional concepts* as T_1 and the extensional type theory for T_2. In this case a full conservativity result actually follows, but the additional extensional requirements on T_1—e.g. the identity proof irrelevance:

$$\lfloor x : A,\ p : x = x \rfloor\ p = \mathsf{r}(x)$$

for paths in between a term and itself—are fundamental. In fact, a dependent type theory with fully intensional identity types does not make the type:

$$\lfloor x, y : A,\ p, q : x = y \rfloor\ p = q : \textsc{Type}$$

inhabited, while the extensional type theory does. This is the main topic of our work.

In this paper we consider dependent type theory T, that we may call *propositional*, seen as a further weakening of a fully intensional dependent type theory, and we look for a concrete family of type judgements such that the extensional type theory is conservative over T *relative* to that family, i.e. whenever the extensional type theory makes one of these types inhabited, so does T—see Theorem 6.6. In detail, the theory T is going to be endowed with *propositional*, or *axiomatic* [OS25, Spa25], *identity types.*

A dependent type theory is said to have *propositional identity types* if it is endowed with a type constructor consisting of the usual formation, introduction, and elimination rules of intensional identity type—see Figure 2—except for the judgemental equality of its computation rule. The latter is only required to hold in a weakened form, called *propositional form*, i.e. it is *replaced by a propositional equality*: whenever we are given judgements:

$$\lfloor x, y : A;\ p : x = y \rfloor\ C(x, y, p) : \textsc{Type} \qquad \text{and} \qquad \lfloor x : A \rfloor\ q(x) : C(x, x, r(x))$$

and hence a term judgement $\lfloor x, y : A;\ p : x = y \rfloor\ \mathsf{J}(q, x, y, p) : C(x, y, p)$ by the elimination rule, then, in place of asking that the term equality judgement $\lfloor x : A \rfloor\ \mathsf{J}(q, x, x, r(x)) \equiv q(x)$ holds, we only ask that it holds *propositionally*, i.e. that an additional term judgement of the form:

$$\lfloor x : A \rfloor\ \mathsf{H}(q, x) : \mathsf{J}(q, x, x, r(x)) = q(x)$$

holds.

Cohen, Coquand, Danielsson, Huber, and Mörtberg [CD13, CCHM18] conducted initial analyses related to propositional identity types. Another work [BCH14] introduces a univalent model of MLTT where the propositional computation rule for identity types is validated, although its judgmental version is not. Following this, the type constructor has been thoroughly examined by van den Berg and Moerdijk [vdB18, vdBM18], who introduced and explored a notion of semantics for dependent type theories with propositional identity types, using the notion of a *path category*. One may consider the same form of weakening for the computation rule of dependent sum types and dependent product types: these type constructors satisfying a propositional computation rule are called *propositional* (or *axiomatic*) *dependent sum types* and *propositional* (or *axiomatic*) *dependent product types* respectively [OS25, Spa25]. The dependent type theory T that we are going to consider is therefore a dependent type theory having propositional identity types, propositional dependent sum types, and propositional dependent product types with *function extensionality* in propositional form, together with an arbitrary family of atomic types and atomic terms. We call such a theory *propositional dependent type theory.* The aim of this paper is to provide a semantic proof that the corresponding extensional type theory is conservative over the propositional type theory T relative to the family of type judgements of the latter obtained by inductively applying the type formation rules to the atomic types that are provably h-sets in the latter.

The property of conservativity between various weakenings of extensional and intensional theories has been studied by several authors, in addition to the already cited Martin Hofmann [Hof96]. Bocquet [Boc20] examines the property of *Morita equivalence* between such a propositional theory and its extension obtained by strictifying the computation rule for identity types. Kapulkin and Li [KL25] reformulate and prove Hofmann's result in terms of such a Morita equivalence. Winterhalter, Sozeau, and Tabareau [WST19], building on Hofmann's point of view and on an approach devised by Oury [Our05], define a translation from extensional theories to intensional ones with uniqueness of identity proofs (UIP) and function extensionality. This translation is then adapted by Boulier and Winterhalter [BW19, Win20] to obtain one from extensional theories to propositional ones, the latter again with uniqueness of identity proofs and function extensionality. This adaptation is used to achieve a completely syntactical proof of the conservativity of the former over the latter.

The present paper can thus be seen as an alternative, semantic, approach to the same problem addressed by Boulier and Winterhalter, building on Hofmann's argument. Our approach involves considering a propositional theory of dependent types, where dependent product types include function extensionality in propositional form, without initially requiring the uniqueness of identity proofs. The argument we follow gradually leads us to restrict the theory to contexts, called *h-elementary*, generated only from h-sets. Consequently, we end up with a result formulated more abstractly than the one contained in [BW19, Win20], with a specific concrete case that essentially matches the latter. This approach also enables us to better understand both the strengths and limitations of Hofmann's semantic argument by starting from a broader perspective. We also emphasise that the inductive notion of h-elementary type—see Definition 5.1—is not merely a technical tool that allows us to continue applying Hofmann's argument, but it is also a natural and expressive family of statements to consider in a theory. For instance, all statements in Heyting arithmetic are h-elementary statements.

Structure of the paper. In Section 2 we recall the notion of category with attributes, that constitutes a notion of semantics for dependent type theories. We recall a well-known notion of *sound* semantics for extensional dependent type theories based on categories with attributes and we present an analogous one for propositional ones.

In Section 3 we consider the notions of homotopy equivalence between types (namely *homotopy equivalence*) and of homotopy equivalence between contexts (*context homotopy equivalence*) in a propositional dependent type theory. These notions identify maps (between type and contexts, respectively) that are invertible up to homotopy (i.e. pointwise propositional equality and pointwise context propositional equality [GG08] respectively). We recall how to extend a homotopy equivalence between two contexts via a homotopy equivalence between two types in the given contexts, and how to obtain new homotopy equivalences between types, by starting from other given ones and applying the type constructors. Finally, we inductively define a family of *canonical* homotopy equivalences and a family of *canonical* context homotopy equivalences, notions appearing in other works [Hof95a, Hof96, Mai09, CM24, MS25] mentioned in Subsection 3.6. This notion is used in Section 5 in order to make the syntax of a "sub-theory" of a propositional type theory into a model of an extensional type theory.

Section 4 is devoted to studying properties of the family of the canonical context homotopy equivalences. These properties are deduced by induction on the complexity of such an equivalence: as the family of the canonical context homotopy equivalences depends on the family of the canonical homotopy equivalences between types, a result for the former usually follows by induction starting from a result for the latter, which is obtained by induction as well. The properties of reflexivity, symmetry and transitivity hold for the general family of the canonical context homotopy equivalences. However, the fundamental property that any two parallel canonical equivalences are homotopic, i.e. pointwise context propositionally equal, does not hold, hence we need to restrict ourselves to the smaller family of canonical equivalences between those contexts that we call *contexts with h-propositional identities* (see Definition 4.14). Having this property is fundamental to ensure the well-definedness of the composition in the category with attributes that we define in Section 5, starting from the syntax of the given propositional type theory. In detail, we use the property of *having h-propositional identities* at the end of the proof of Lemma 4.18.

Section 5 defines a category with attributes $\mathfrak{C}$ starting from the *h-elementary* contexts of the given propositional type theory, identified up to canonical context homotopy equivalence. The h-elementary contexts are a special case of contexts with h-propositional identities that we define at the beginning of the section (see Definition 5.2). This restriction makes the natural family of display maps that we define for $\mathfrak{C}$ into a *cartesian* natural transformation, so that we obtain a model of the strict substitution. Specifically, the h-elementariness property is used to deduce the uniqueness of the factorisation of a given square against a naturality square. We conclude our analysis by proving in Section 6 that $\mathfrak{C}$ is actually a model of the extensional type theory. This allows us to infer, by soundness, the conservativity of the extensional theory over the h-elementary contexts of the propositional one.

1.1. **Preliminary conventions and terminology.** In this paper, a dependent type theory is intended to be *extensional* if it has extensional identity types, dependent product types, and dependent sum types, i.e. if it satisfies the rules of Figure 1, Figure 4, and Figure 6 (see Section 7) respectively—or, for dependent sums, their equivalent formulation in Figure 8. Analogously, a dependent type theory is said to be *propositional* if it has propositional identity types, propositional dependent product types with the propositional function extensionality rule, and propositional dependent sum types, i.e. if it satisfies the rules of Figure 3, Figure 5, and Figure 7 respectively—or, for propositional dependent sums, their equivalent formulation in Figure 9.

However, for sake of simplicity, for the remainder of the paper we consider *one* propositional type theory: i.e. we assume that we are given a dependent type theory, which is propositional, together with a family of atomic types and atomic terms and subject to the usual structural rules—for an enumeration see [Str91, Chapter III—context formation, context equality, judgement formation, judgement equality], [Jac99], [Hof97]. We refer to this specific dependent type theory as Propositional Type Theory (PTT). Analogously, as there is only one extensional type theory that we actually consider in this paper (relative to the given PTT), i.e. the one whose atomic types are the atomic types of PTT that are provably *h-sets* in PTT (and whose atomic terms are the atomic terms of these atomic h-sets in PTT), we refer to it as Extensional Type Theory (ETT).

We use the symbology $\lfloor - \rfloor$ $-$ to indicate judgements of PTT and the symbology $\lfloor - \rfloor_{\mathtt{ext}}$ $-$ to indicate the ones of ETT. We use the symbol $\equiv$ to indicate judgmental equalities between contexts, types in the same context and terms in the same context and of the same type. We use the symbol $=$ to indicate propositional equalities i.e. identity types. Sometimes (especially in diagrams) we adopt the notations $x \Rightarrow y$ and $x \Leftarrow y$, where $x \Rightarrow y$ indicates the type $x = y$ and $x \Leftarrow y$ indicates the type $y = x$.

Sometimes, we use path induction on paths that may not appear to be general, i.e. paths of the form $x, y : A, p : x = y$ for some type A. However, in these instances, we mean that the specific type on which we are performing path induction can be generalised so that the specific path p' is replaced by a general path as p. This allows us to use path induction, after which we substitute p' for the general path p in order to obtain the desired result. This argument is indicated as *generalised path induction* or *generalised path elimination*.

2. Recap on categories with attributes

This section is devoted to the notion of a *category with attributes* and its use to define an opportune notion of semantics for several kinds of dependent type theories. For further details, we refer the reader to [Car78, Mog91, KL21].

Definition 2.1 (Category with attributes)**.** Suppose that we are given:

- A category $\mathcal{C}$ with terminal object 1, whose objects are called *(semantic) contexts*.
- A functor $\mathcal{C}^{\mathrm{op}} \xrightarrow{\textsc{tp}} \textsc{Discr}$, that we call *presheaf of (semantics) types*. If Γ is a semantic context, then an object A of the category $\textsc{tp}\Gamma$ is said to be a *semantic type* in semantic context Γ. Here $\textsc{Discr}$ denotes the category of (small) discrete categories.
- A functor $\int \textsc{tp} \xrightarrow{-.-} \mathcal{C}$, that we call *(semantic) context extension*. Here $\int \textsc{tp}$ denotes the Grothendieck construction associated to the presheaf $\textsc{tp}$.
- A cartesian natural transformation:
$$\Big(\int \textsc{tp} \xrightarrow{-.-} \mathcal{C}\Big) \xrightarrow{P} \Big(\int \textsc{tp} \xrightarrow{\pi} \mathcal{C}\Big)$$
where π denotes the projection on the first component (we remind that a natural transformation is said to be *cartesian* when its naturality squares are pullbacks). The (Γ, A)-component:
$$\Gamma.A \xrightarrow{P_A} \Gamma$$
of P is called *display map* of A.

Then we say that the quadruple $(\mathcal{C}, \textsc{tp}, -.-, P)$ is a *category with attributes*.

The notion of category with attributes constitutes a notion of semantics for dependent type theories with the usual structural rules [Jac99, Hof97][Str91, Chapter III—context formation, context equality, judgement formation, judgement equality], hence with the usual *strict* notion of substitution. In this case, semantic terms of a category with attributes are presented as sections of the display maps. A modification of this concept, producing an equivalent notion of semantics, is used by other authors [CD14, Dyb96, Hof95a, Hof97], where one requires that, for every semantic type A, a set of *semantic terms* is given. Other requirements imply that this set is in bijection with the sections of the corresponding display map P_A. In our case (see Remark 2.2), the semantic terms of A are defined to be the the sections of P_A themselves: it turns out that our notion of category with attributes is the one of [Hof97] in the special case where the bijections between the semantic terms (of given semantic types) and the sections (of the corresponding display maps) are identities.

For an actually more general notion of semantics for dependent type theories we refer the reader to the notion of *comprehension category*, see [Jac93], or to the one of *display map category*, see [Tay99, Jac99, MvG18]. In this case, in fact, a dependent type theory modelled by such a structure does not necessarily enjoy a strictly functorial notion of substitution, analogously to the case of locally cartesian closed categories (see [See84]) as pointed out by Hofmann [Hof95b]. For more details on a pseudo-functorial notion of substitution in dependent type theories, we refer the reader to [Cur93].

Remark 2.2 (Notation and terminology)**.** Let $(\mathcal{C}, \textsc{tp}, -.-, P)$ be a category with attributes.

- For the (Γ, A)-component of P, we do not explicitly write the dependence on Γ. This does not generate ambiguity because, whenever we consider a semantic type, its semantic context is always specified.
- If f is a morphism of semantic contexts $\Delta \to \Gamma$, then we denote as $-[f]$ the action:
$$\textsc{tp}\Gamma \to \textsc{tp}\Delta$$
of $\textsc{tp}$ on f. Hence, if A is a semantic type in Γ, then $A[f]$ denotes the action of $-[f]$ on A. However, for sake of clarity we sometimes use the general notation $\textsc{tp}f$ in place of $-[f]$, and hence we write $(\textsc{tp}f)A$ in place of $A[f]$. E.g. this happens in Section 5 and Section 6.
- We remind that the objects of $\int \textsc{tp}$ are pairs (Γ, A), where Γ is a semantic context and A is semantic type in Γ. An arrow $(\Delta, B) \to (\Gamma, A)$ of $\int \textsc{tp}$ is a pair (f, φ), where f is an arrow $\Delta \to \Gamma$ and φ an arrow $B \to A[f]$. We refer to [MR13] for further details on this notion. The category $\textsc{tp}\Delta$ is discrete, therefore an arrow $(\Delta, B) \to (\Gamma, A)$ is nothing but a morphism of contexts $\Delta \to \Gamma$ such that B is the type $A[f]$ in Δ. As we got rid of the dependence on the φ component, we denote the image of such an arrow via the functor $-.-$ as:
$$\Delta.A[f] \xrightarrow{f.A} \Gamma.A$$
and any naturality (pullback, since P is cartesian) square of P is of the form:
$$\begin{array}{ccc} \Delta.A[f] & \xrightarrow{f.A} & \Gamma.A \\ {\scriptstyle P_{A[f]}}\downarrow & & \downarrow{\scriptstyle P_A} \\ \Delta & \xrightarrow{f} & \Gamma \end{array}$$
for some semantic context morphism f.
- Let A be a semantic type in semantic context Γ. The sections $\Gamma \xrightarrow{a} \Gamma.A$ of its display map $\Gamma.A \xrightarrow{P_A} \Gamma$ are called *(semantic) terms* of A in Γ.

In a given dependent type theory, whenever we are given contexts $\gamma : \Gamma$ and $\delta : \Delta$, a type $\lfloor \delta : \Delta \rfloor\ B(\delta) : \textsc{Type}$ and a morphism of contexts $\lfloor \gamma : \Gamma \rfloor\ f(\gamma) : \Delta, B$ (see the beginning of Section 3 for both the notion of *morphism of contexts* and the meaning of this notation), then we implicitly mean that we are given a judgement of the form:

$$\lfloor \gamma : \Gamma \rfloor \ \overline{f}(\gamma) : B(P_{B(\delta)} f(\gamma))$$

such that the substitution $f(\gamma) : \Delta, B(\delta)$ decomposes as:

$$P_{B(\delta)} f(\gamma) : \Delta, \ \overline{f}(\gamma) : B(P_{B(\delta)} f(\gamma))$$

where $P_{B(\delta)}$ denotes the substitution $\lfloor \delta : \Delta, y : B(\delta) \rfloor \ \delta : \Delta$. Every category with attributes models this phenomenon:

Proposition 2.3. *Let $(\mathcal{C}, \text{TP}, -.-, P)$ be a category with attributes. Let f be a semantic context morphism $\Gamma \to \Delta.B$, where B is a semantic type in Δ. Then there is a unique semantic term $\overline{f}$ of $B[P_B f]$ in Γ such that the diagram:*

$$\begin{array}{ccccc} \Gamma & & \xrightarrow{\quad f \quad} & & \Delta.B \\ & \searrow^{\overline{f}} & & \nearrow_{P_B f.B} & \\ & & \Gamma.B[P_B f] & & \end{array}$$

commutes. Moreover, if f is of the form:

$$\Gamma \xrightarrow{b} \Gamma.B[g] \xrightarrow{g.B} \Delta.B$$

for some semantic context morphism $\Gamma \xrightarrow{g} \Delta$ and some semantic term b of $B[g]$, then the semantic term $\overline{f}$ is b itself (observe that:

$$P_B f = P_B (g.B) b = g P_{B[g]} b = g$$

hence in this case $\Gamma.B[P_B f] = \Gamma.B[g]$, so that the equality $\overline{f} = b$ typechecks).

Proof. By cartesianity of P. □

2.1. **Extending substitution to terms.** In a category with attributes $(\mathcal{C}, \text{TP}, -.-, P)$ the notion of substitution for semantic types (given by the presheaf TP) naturally extends to semantic terms. Suppose that we are given a morphism of semantic contexts $\Delta \xrightarrow{f} \Gamma$, a semantic type A in Γ and a semantic term a of A. As the square:

$$\begin{array}{ccc} \Delta.A[f] & \xrightarrow{f.A} & \Gamma.A \\ {\scriptstyle P_{A[f]}}\downarrow & & \downarrow{\scriptstyle P_A} \\ \Delta & \xrightarrow{f} & \Gamma \end{array}$$

is a pullback and $P_A a f = f$, there is a unique section $a[f]$ of $P_{A[f]}$ such that:

$$\begin{array}{ccc} \Delta & \xrightarrow{f} & \Gamma \\ {\scriptstyle a[f]}\downarrow & & \downarrow{\scriptstyle a} \\ \Delta.A[f] & \xrightarrow{f.A} & \Gamma.A \end{array}$$

commutes. We define the action of the substitution $-[f]$ on a as the semantic term $a[f]$ of $A[f]$. As expected:

Proposition 2.4. *For a semantic term a of type A in context Γ and a semantic morphism $\Delta \xrightarrow{f} \Gamma$, the semantic terms $a[f]$ and $\overline{af}$ coincide.*

Proof. It holds that $af = (f.A)a[f]$. Hence by Proposition 2.3 $\overline{af}$ is $a[f]$. □

and additionally:

Proposition 2.5. *For a semantic term a, the semantic term $a[f]$ is functorial in f.*

Proof. The term $a[1_\Gamma]$ of $A[1_\Gamma] = A$ makes the diagram:

$$\begin{array}{ccc} \Gamma & \xrightarrow{1_\Gamma} & \Gamma \\ \big\downarrow{\scriptstyle a[1_\Gamma]} & & \big\downarrow{\scriptstyle a} \\ \Gamma.A & \xrightarrow{1_\Gamma.A = 1_{\Gamma.A}} & \Gamma.A \end{array}$$

commute, hence $a[1_\Gamma] = a$.

If we are given morphisms of contexts $\Omega \xrightarrow{g} \Delta \xrightarrow{f} \Gamma$ then the minimal squares in the diagram:

$$\begin{array}{ccccc} \Omega & \xrightarrow{g} & \Delta & \xrightarrow{f} & \Gamma \\ \big\downarrow{\scriptstyle a[f][g]} & & \big\downarrow{\scriptstyle a[f]} & & \big\downarrow{\scriptstyle a} \\ \Omega.A[f][g] & \xrightarrow{g.A[f]} & \Delta.A[f] & \xrightarrow{f.A} & \Gamma.A \end{array}$$

commute, hence the outer rectangle—whose lower side is $(f.A)(g.A[f]) = (fg).A$—commutes as well. As $a[fg]$ is the unique term of $A[fg] = A[f][g]$ making the outer rectangle commute, we conclude that $a[f][g] = a[fg]$. □

We end the current subsection recalling the following:

Lemma 2.6. *Suppose that a and b are semantic terms of some semantic type A in some semantic context Γ. Then:*

$$a;b := (\ \Gamma \xrightarrow{b} \Gamma.A \xrightarrow{a.A[P_A]} \Gamma.A.A[P_A]\ ) = (\ \Gamma \xrightarrow{a} \Gamma.A \xrightarrow{b[P_A]} \Gamma.A.A[P_A]\ )$$

and $a;b$ is the unique arrow $\Gamma \to \Gamma.A.A[P_A]$ whose postcompositions via $P_{A[P_A]}$ and via $P_A.A$ are a and b respectively. Moreover, if $\Delta \xrightarrow{f} \Gamma$ is a morphism of contexts, then:

$$\begin{array}{ccccc} \Delta & & \xrightarrow{\qquad\qquad f \qquad\qquad} & & \Gamma \\ \big\downarrow{\scriptstyle a[f];b[f]} & & & & \big\downarrow{\scriptstyle a;b} \\ \Delta.A[f].A[f][P_{A[f]}] & = & \Delta.A[f].A[P_A(f.A)] & \xrightarrow{f.A.A[P_A]} & \Gamma.A.A[P_A] \end{array}$$

commutes.

Proof. By building the term $b = b[P_A a]$ according to the definition of $b[P_A a]$ itself:

$$\begin{array}{ccccc}
\Gamma & \xrightarrow{\quad a \quad} & \Gamma.A & \xrightarrow{\quad P_A \quad} & \Gamma \\
\big\downarrow{\scriptstyle b[P_A a]} & & \big\downarrow{\scriptstyle b[P_A]} & & \big\downarrow{\scriptstyle b} \\
\Gamma.A = \Gamma.A[P_A a] & \xrightarrow{a.A[P_A]} & \Gamma.A.A[P_A] & \xrightarrow{P_A.A} & \Gamma.A \\
\big\downarrow{\scriptstyle P_A = P_{A[P_A a]}} & & \big\downarrow{\scriptstyle P_{A[P_A]}} & & \big\downarrow{\scriptstyle P_A} \\
\Gamma & \xrightarrow{\quad a \quad} & \Gamma.A & \xrightarrow{\quad P_A \quad} & \Gamma
\end{array}$$

the upper left-hand square commutes, hence we are done with the first equality. Since $P_{A[P_A]} b[P_A] a = a$ and $(P_A.A)(a.A[P_A])b = ((P_A a).A)b = b$, we conclude that $a; b$ is the unique arrow $\Gamma \to \Gamma.A.A[P_A]$ whose postcompositions via $P_{A[P_A]}$ and via $P_A.A$ are a and b respectively.

In order to verify that the diagram of the statement commutes, we use the first presentation:

$$\Delta \xrightarrow{b[f]} \Delta.A[f] \xrightarrow{a[f].A[f][P_{A[f]}]} \Delta.A[f].A[f][P_{A[f]}]$$

for $a[f]; b[f]$, and the second one:

$$\Gamma \xrightarrow{a} \Gamma.A \xrightarrow{b[P_A]} \Gamma.A.A[P_A]$$

for $a; b$. We are left to verify that:

$$\begin{array}{ccccccc}
\Delta & & \xrightarrow{\qquad\qquad f \qquad\qquad} & & \Gamma & \xrightarrow{\quad a \quad} & \Gamma.A \\
\big\downarrow{\scriptstyle b[P_A(af)]} & & & & & & \big\downarrow{\scriptstyle b[P_A]} \\
\Delta.A[f] & \xrightarrow{a[f].A[f][P_{A[f]}]} & \Delta.A[f].A[f][P_{A[f]}] & = & \Delta.A[f].A[P_A(f.A)] & \xrightarrow{f.A.A[P_A]} & \Gamma.A.A[P_A]
\end{array}$$

commutes, since $b[P_A(af)] = b[f]$. Since the diagram:

$$\begin{array}{ccccc}
\Delta & \xrightarrow{\quad f \quad} & \Gamma & \xrightarrow{\quad a \quad} & \Gamma.A \\
\big\downarrow{\scriptstyle b[P_A(af)]} & & & & \big\downarrow{\scriptstyle b[P_A]} \\
\Delta.A[f] & \xrightarrow{\quad f.A \quad} & \Gamma.A & \xrightarrow{a.A[P_A]} & \Gamma.A.A[P_A]
\end{array}$$

commutes (and this is true because the equality $(a.A[P_A])(f.A) = (af).A[P_A]$ holds) we are done if we verify that:

$$\begin{array}{ccc}
\Delta.A[f] & \xrightarrow{\qquad f.A \qquad} & \Gamma.A \\
\big\downarrow{\scriptstyle a[f].A[P_A(f.A)]} & & \big\downarrow{\scriptstyle a.A[P_A]} \\
\Delta.A[f].A[P_A(f.A)] & \xrightarrow{\quad f.A.A[P_A] \quad} & \Gamma.A.A[P_A]
\end{array}$$

commutes and use that $a[f].A[f][P_{A[f]}] = a[f].A[P_A(f.A)]$. This is actually true as:

$$(\ f.A.A[P_A]\ )(\ a[f].A[P_A(f.A)]\ ) = (\ (f.A)a[f]\ ).A[P_A] = (\ af\ ).A[P_A] = (\ a.A[P_A]\ )(\ f.A\ )$$

hence we are done. □

Remark 2.7. If we are given a dependent type theory, if we are given judgements of the form $\lfloor \gamma \rfloor\ a(\gamma) : A(\gamma)$ and $\lfloor \gamma \rfloor\ b(\gamma) : A(\gamma)$, then the morphism of semantic contexts $a; b$ of Lemma 2.6 corresponds to the morphism of contexts $\lfloor \gamma \rfloor\ \gamma, a(\gamma), b(\gamma)$.

2.2. **Variable terms.** Let $(\mathcal{C}, \text{TP}, -.-, P)$ be a category with attributes and let A be a semantic type in a semantic context Γ. Then, we might consider the semantic term $\overline{1_{\Gamma.A}}$ of $A[P_A]$, that is a section:

$$\Gamma.A \xrightarrow{\overline{1_{\Gamma.A}}} \Gamma.A.A[P_A]$$

of $P_{A[P_A]}$. We denote this semantic term as v_A and call it *semantic variable* of A and remind that it is characterised, among the sections of $P_{A[P_A]}$, as the one satisfying $(P_A.A)v_A = 1_{\Gamma.A}$. We recall, without proof, some important equalities:

Lemma 2.8. *Let $\Delta \xrightarrow{f} \Gamma$ be an arrow in $\mathcal{C}$ and consider the corresponding extension $\Delta.A[f] \xrightarrow{f.A} \Gamma.A$. Then the equality:*

$$v_A[f.A] = v_{[A[f]]}$$

between semantic terms of $A[P_A(f.A)] = A[fP_{A[f]}]$ holds.

Lemma 2.9. *If a is a semantic term of A, then:*

$$(\ \Gamma \xrightarrow{a} \Gamma.A \xrightarrow{v_A} \Gamma.A.A[P_A]\ ) = (\ \Gamma \xrightarrow{a} \Gamma.A \xrightarrow{a.A[P_A]} \Gamma.A.A[P_A]\ ).$$

In particular, the equality $v_A[a] = a$ between sections of P_A holds.

Remark 2.10. In a given dependent type theory, if $\gamma : \Gamma$ is a context and $\lfloor \gamma \rfloor\ A(\gamma) : \text{Type}$ is a type, then two morphisms of contexts naturally arise:

$$\lfloor \gamma, x : A(\gamma) \rfloor\ \gamma : \Gamma$$
$$\lfloor \gamma, x : A(\gamma) \rfloor\ x : A(\gamma) \equiv A(P_{A(\gamma)}(\gamma, x)).$$

The former corresponds to the morphism of contexts P_A in a given category with attributes, while the latter, seen as a term rather then a general morphism of contexts, to the semantic term v_A associated to a given semantic type A.

2.3. **Semantics of extensional and propositional type theories.** In this subsection we briefly describe the additional structure that a category with attributes needs to be equipped with in order to allow on it an interpretation of an extensional type theory or of a propositional one.

The following two notions, Definition 2.11 and Definition 2.12, define when a category with attributes is equipped with semantic identity types (in the extensional and propositional case, respectively). In detail, the following notion is meant to model the inference rules for the extensional identity types, that here we recall in a concise form:

$$\text{Form}\ \frac{A : \textsc{Type}}{\lfloor x, x' : A \rfloor\ x = x' : \textsc{Type}} \qquad \text{Extensionality}\ \frac{A : \textsc{Type}}{\lfloor x, x' : A;\ p : x = x' \rfloor\ x \equiv x'}$$

$$\text{Intro}\ \frac{A : \textsc{Type}}{\lfloor x : A \rfloor\ \mathsf{r}(x) : x = x} \qquad \text{Id proof irrelevence}\ \frac{A : \textsc{Type}}{\lfloor x, x' : A;\ p : x = x' \rfloor\ p \equiv \mathsf{r}(x)}$$

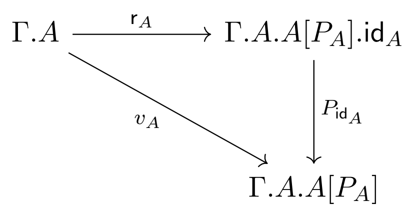
referring the reader to Figure 1 for the extended version.

Definition 2.11 (Semantic extensional identity types)**.** We say that a category with attributes $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **equipped with semantic extensional identity types** if, for every semantic context Γ and every semantic type A in context Γ, there is a choice of:

- (*Formation*) a semantic type id_A of semantic context $\Gamma.A.A[P_A]$;
- (*Introduction*) a morphism of contexts:

$$\Gamma.A \xrightarrow{\mathsf{r}_A} \Gamma.A.A[P_A].\mathsf{id}_A$$

  such that the diagram:



  commutes;

in such a way that the following properties are satisfied:

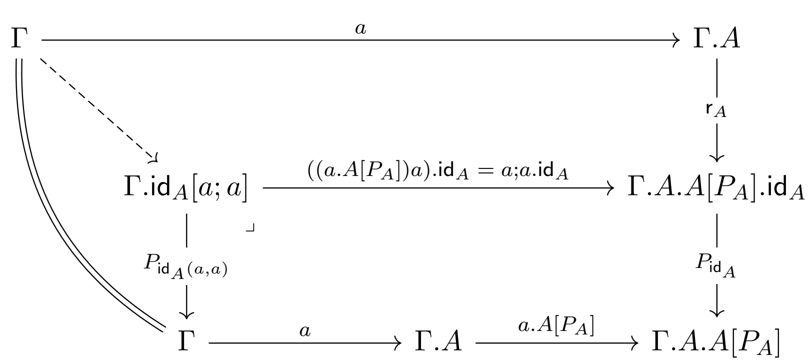
- (*Extensionality*) For all semantic terms a and b of semantic type A and every semantic term $\Gamma \xrightarrow{p} \Gamma.\mathsf{id}_A(a, b)$ of semantic type:

$$\mathsf{id}_A[a; b] = \mathsf{id}_A[\ \Gamma \xrightarrow{b} \Gamma.A \xrightarrow{a.A[P_A]} \Gamma.A.A[P_A]\ ] = \mathsf{id}_A[\ \Gamma \xrightarrow{a} \Gamma.A \xrightarrow{b[P_A]} \Gamma.A.A[P_A]\ ]$$

  in context Γ, the equality $a = b$ between semantic terms of type A and the equality:

$$(\ \Gamma \xrightarrow{p} \Gamma.\mathsf{id}_A[a; b] = \Gamma.\mathsf{id}_A[a; a]\ ) = (\ \Gamma \xrightarrow{\mathsf{r}^a_A} \Gamma.\mathsf{id}_A[a; a]\ )$$

  between semantic terms of type $\mathsf{id}_A[a; a]$ hold. Here r^a_A is the unique arrow $\Gamma \dashrightarrow \Gamma.\mathsf{id}_A[a; a]$ such that:



  commutes (observe that $P_{\mathsf{id}_A}\mathsf{r}_A a = v_A a = (a.A[P_A])a$ by Lemma 2.9).

- (*Compatibility with the substitution*) If f is a morphism of semantic contexts $\Delta \to \Gamma$ then the equality:
$$\mathsf{id}_A[\ \Delta.A[f].A[fP_{A[f]}] \xrightarrow{f.A.A[P_A]} \Gamma.A.A[P_A]\ ] = \mathsf{id}_{A[f]}$$
between semantic types in context $\Delta.A[f].A[fP_{A[f]}]$ holds and:
$$\mathsf{r}_A[f] = \mathsf{r}_{A[f]}$$
where $\mathsf{r}_A[f]$ is the unique arrow $\Delta.A[f] \to \Delta.A[f].A[f][P_{A[f]}].\mathsf{id}_{A[f]}$ such that $P_{\mathsf{id}_A}\mathsf{r}_A[f] = v_A$ and such that the diagram:
$$\begin{array}{ccc} \Delta.A[f] & \xrightarrow{\quad f.A \quad} & \Gamma.A \\ \downarrow{\scriptstyle \mathsf{r}_A[f]} & & \downarrow{\scriptstyle \mathsf{r}_A} \\ \Delta.A[f].A[f][P_{A[f]}].\mathsf{id}_{A[f]} & \xrightarrow{f.A.A[P_A].\mathsf{id}_A} & \Gamma.A.A[P_A].\mathsf{id}_A \end{array}$$
commutes.

The following notion is meant to model the inference rules for the propositional identity types, that here we recall in a concise form:

$$\text{Form}\ \frac{A : \textsc{Type}}{\lfloor x, x' : A \rfloor\ x = x' : \textsc{Type}} \qquad \text{Elim}\ \frac{\begin{array}{c} A : \textsc{Type} \\ \lfloor x, x';\ p : x = x' \rfloor\ C(x, x', p) : \textsc{Type} \\ \lfloor x \rfloor\ c(x) : C(x, x, \mathsf{r}(x)) \end{array}}{\lfloor x, x';\ p \rfloor\ \mathsf{J}(c, x, x', p) : C(x, x', p)}$$

$$\text{Intro}\ \frac{A : \textsc{Type}}{\lfloor x : A \rfloor\ \mathsf{r}(x) : x = x} \qquad \text{Prop comp}\ \frac{\begin{array}{c} A : \textsc{Type} \\ \lfloor x, x';\ p : x = x' \rfloor\ C(x, x', p) : \textsc{Type} \\ \lfloor x \rfloor\ c(x) : C(x, x, \mathsf{r}(x)) \end{array}}{\lfloor x \rfloor\ \mathsf{H}(c, x) : \mathsf{J}(c, x, x, \mathsf{r}(x)) = c(x)}$$

referring the reader to Figure Figure 3 for the extended version.

Definition 2.12 (Semantic propositional identity types)**.** We say that a category with attributes $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **equipped with semantic propositional identity types** if it satisfies *formation*, *introduction*, *compatibility with the substitution* of Definition 2.11 and moreover:

- (*Elimination and propositional computation*) for every semantic context Γ, every semantic type A in context Γ, every semantic type C in context $\Gamma.A.A[P_A].\mathsf{id}_A$ and every semantic term $\Gamma.A \xrightarrow{c} \Gamma.A.C[\mathsf{r}_A]$ of $C[\mathsf{r}_A]$ in context $\Gamma.A$, there is a choice of a semantic term:
$$\Gamma.A.A[P_A].\mathsf{id}_A \xrightarrow{\mathsf{J}_c} \Gamma.A.A[P_A].\mathsf{id}_A.C$$
of type C in context $\Gamma.A.A[P_A].\mathsf{id}_A$ and of a semantic term:
$$\Gamma.A \xrightarrow{\mathsf{H}_c} \Gamma.A.\mathsf{id}_{C[\mathsf{r}_A]}[\mathsf{J}_c[\mathsf{r}_A]; c]$$
of type $\mathsf{id}_{C[\mathsf{r}_A]}[\mathsf{J}_c[\mathsf{r}_A]; c]$ in context $\Gamma.A$;

- (*Additional compatibility with the substitution*) for every semantic context Γ, every semantic type A in context Γ, every semantic type C in context $\Gamma.A.A[P_A].\mathsf{id}_A$, every semantic term $\Gamma.A \xrightarrow{c} \Gamma.A.C[\mathsf{r}_A]$ of $C[\mathsf{r}_A]$ in context $\Gamma.A$ and every morphism of semantic contexts $\Delta \xrightarrow{f} \Gamma$, the diagram:

$$\begin{array}{ccc}
\Delta.A[f].A[fP_{A[f]}].\mathsf{id}_{A[f]} & = & \Delta.A[f].A[P_A(f.A)].\mathsf{id}_A[f.A.A[P_A]] \\
\Big\downarrow \mathsf{J}_{\left(\Delta.A[f] \xrightarrow{c[f.A]} \Delta.A[f].C[\mathsf{r}_A(f.A)]=\Delta.A[f].C[(f.A.A[P_A].\mathsf{id}_A)\mathsf{r}_{A[f]}]\right)} & & \Big\downarrow \mathsf{J}_c[f.A.A[P_A].\mathsf{id}_A] \\
\Delta.A[f].A[fP_{A[f]}].\mathsf{id}_{A[f]}.C' & = & \Delta.A[f].A[P_A(f.A)].\mathsf{id}_A[f.A.A[P_A]].C'
\end{array}$$

where $C' \equiv C[f.A.A[P_A].\mathsf{id}_A]$, commutes i.e. the equality $\mathsf{J}_{c[f.A]} = \mathsf{J}_c[f.A.A[P_A].\mathsf{id}_A]$ holds, and that the diagram:

$$\begin{array}{ccc}
\Delta.A[f] & \xrightarrow{\mathsf{H}_{c[f.A]}} & \Delta.A[f].\mathsf{id}_{C[(f.A.A[P_A].\mathsf{id}_A)\mathsf{r}_{A[f]}]}[\ \mathsf{J}_{c[f.A]}[\mathsf{r}_{A[f]}]; c[f.A]\ ] \\
\Vert & & \Vert \\
 & & \Delta.A[f].\mathsf{id}_{C[\mathsf{r}_A][f.A]}[\ \mathsf{J}_c[f.A.A[P_A].\mathsf{id}_A][\mathsf{r}_{A[f]}]; c[f.A]\ ] \\
 & & \Vert \\
 & & \Delta.A[f].\mathsf{id}_{C[\mathsf{r}_A][f.A]}[\ \mathsf{J}_c[\mathsf{r}_A][f.A]; c[f.A]\ ] \\
 & & \Vert \\
 & & \Delta.A[f].(\ \mathsf{id}_{C[\mathsf{r}_A]}[\ f.A.C[\mathsf{r}_A].C[\mathsf{r}_A P_{C[\mathsf{r}_A]}]\ ]\ )[\ \mathsf{J}_c[\mathsf{r}_A][f.A]; c[f.A]\ ] \\
 & & \Vert \text{ Lemma 2.6} \\
\Delta.A[f] & \xrightarrow[\mathsf{H}_c[f.A]]{} & \Delta.A[f].(\ \mathsf{id}_{C[\mathsf{r}_A]}[\mathsf{J}_c[\mathsf{r}_A]; c]\ )[f.A]
\end{array}$$

commutes i.e. the equality $\mathsf{H}_{c[f.A]} = \mathsf{H}_c[f.A]$ holds.

The following two notions (Definition 2.13 and Definition 2.14) define when a category with attributes is equipped with semantic dependent product types (in the extensional and propositional case, respectively). In detail, the following notion is meant to model the inference rules for the dependent product types, that here we recall in a concise form:

$$\text{Form}\ \frac{A : \text{Type} \qquad \lfloor x : A \rfloor\ B(x) : \text{Type}}{\Pi_{x:A} B(x) : \text{Type}} \qquad \text{Intro}\ \frac{\begin{array}{c} A : \text{Type} \qquad \lfloor x : A \rfloor\ B(x) : \text{Type} \\ \lfloor x : A \rfloor\ y(x) : B(x) \end{array}}{\lambda x.y(x) : \Pi_{x:A} B(x)}$$

$$\text{Elim}\ \frac{A : \text{Type} \qquad \lfloor x : A \rfloor\ B(x) : \text{Type}}{\lfloor z : \Pi_{x:A} B(x);\ x : A \rfloor\ \mathsf{ev}(z, x) : B(x)} \qquad \text{Comp}\ \frac{\begin{array}{c} A : \text{Type} \qquad \lfloor x : A \rfloor\ B(x) : \text{Type} \\ \lfloor x : A \rfloor\ y(x) : B(x) \end{array}}{\lfloor x : A \rfloor\ \mathsf{ev}(\lambda x.y(x), x) \equiv y(x)}$$

referring the reader to Figure 4 for the extended version. Here, we also present the semantic counterpart of the expansion rule, which follows automatically in the presence of extensional identity types.

Definition 2.13 (Semantic dependent product types)**.** We say that a category with attributes $(\mathcal{C}, \text{TP}, -.-, P)$ is **equipped with semantic dependent product types** if:

- (*Formation*) for every semantic context Γ, every semantic type A in context Γ and every semantic type B in context $\Gamma.A$, there is a choice of a semantic type $\mathbf{\Pi}_A^B$;
- (*Introduction*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$ and every semantic term $\Gamma.A \xrightarrow{b} \Gamma.A.B$ of B, there is a choice of a semantic term $\Gamma \xrightarrow{\lambda b} \Gamma.\mathbf{\Pi}_A^B$ of $\mathbf{\Pi}_A^B$;
- (*Elimination*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic term $\Gamma \xrightarrow{z} \Gamma.\mathbf{\Pi}_A^B$ of $\mathbf{\Pi}_A^B$ and every semantic term $\Gamma \xrightarrow{a} \Gamma.A$ of A, a choice of a semantic term $\Gamma \xrightarrow{\mathsf{ev}_z^a} \Gamma.B[a]$ of $B[a]$;

in such a way that the following properties are satisfied:

- (*Compatibility with the substitution*) For every semantic context Γ, every semantic type A in context Γ and every semantic type B in context $\Gamma.A$, and for every choice of a semantic term $\Gamma \xrightarrow{z} \Gamma.\mathbf{\Pi}_A^B$ of $\mathbf{\Pi}_A^B$, of a semantic term $\Gamma \xrightarrow{a} \Gamma.A$ of A and of a semantic term $\Gamma.A \xrightarrow{b} \Gamma.A.B$ of B, if $\Delta \xrightarrow{f} \Gamma$ is a morphism of semantic contexts, then:
  - the equality:
  $$\mathbf{\Pi}_A^B[f] = \mathbf{\Pi}_{A[f]}^{B[f.A]}$$
  between semantic types in context Γ holds;
  - the equality:
  $$\left(\mathsf{ev}_{\Gamma \xrightarrow{z} \Gamma.\mathbf{\Pi}_A^B}^{\Gamma \xrightarrow{a} \Gamma.A}\right)[f] = \left(\mathsf{ev}_{\Gamma \xrightarrow{z[f]} \Gamma.\mathbf{\Pi}_{A[f]}^{B[f.A]}}^{\Gamma \xrightarrow{a[f]} \Gamma.A[f]}\right)$$
  between semantic terms of type $B[\ \Delta \xrightarrow{f} \Gamma \xrightarrow{a} \Gamma.A\ ] = B[\ \Delta \xrightarrow{a[f]} \Delta.A[f] \xrightarrow{f.A} \Gamma.A\ ]$ holds;
  - the equality:
  $$(\ \lambda(\ \Gamma.A \xrightarrow{b} \Gamma.A.B\ )\ )[f] = \lambda(\ \Delta.A[f] \xrightarrow{b[f.A]} \Delta.A[f].B[f.A]\ )$$
  between semantic terms of type $\mathbf{\Pi}_A^B[f] = \mathbf{\Pi}_{A[f]}^{B[f.A]}$ holds.
- (*Computation*) If $\Gamma.A \xrightarrow{b} \Gamma.A.B$ is a semantic term of B and if $\Gamma \xrightarrow{a} \Gamma.A$ is a semantic term of A, for some semantic context Γ, some semantic type A in context Γ and some semantic type B in context $\Gamma.A$, then the equality:
  $$(\ \Gamma \xrightarrow{\mathsf{ev}_{\lambda b}^a} \Gamma.B[a]\ ) = (\ \Gamma \xrightarrow{b[a]} \Gamma.B[a]\ )$$
  between semantic terms of type $B[a]$ holds.
- (*Expansion*) If $\Gamma \xrightarrow{z} \Gamma.\mathbf{\Pi}_A^B$ is a semantic term of type $\mathbf{\Pi}_A^B$, then the equality:
  $$\lambda(\ \Gamma.A \xrightarrow{\left(\mathsf{ev}_{\left(\ \Gamma.A \xrightarrow{z[P_A]} \Gamma.A.\mathbf{\Pi}_A^B[P_A] = \Gamma.A.\mathbf{\Pi}_{A[P_A]}^{B[P_A.A]}\ \right)}^{\left(\ \Gamma.A \xrightarrow{v_A} \Gamma.A.A[P_A]\ \right)}\right)} \Gamma.A.B[\ (P_A.A)v_A\ ] = \Gamma.A.B\ ) = z$$
  between semantic terms of type $\mathbf{\Pi}_A^B$ holds.

The following notion is meant to model the inference rules for the propositional dependent product types, that here we recall in a concise form:

$$\text{Form}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\Pi_{x:A} B(x) : \textsc{Type}} \qquad \text{Intro}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \quad \lfloor x : A \rfloor\ y(x) : B(x)}{\lambda x.y(x) : \Pi_{x:A} B(x)}$$

$$\text{Elim}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\lfloor z : \Pi_{x:A} B(x);\ x : A \rfloor\ \mathsf{ev}(z, x) : B(x)} \qquad \text{Prop comp}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \quad \lfloor x : A \rfloor\ y(x) : B(x)}{\lfloor x : A \rfloor\ \beta^{\Pi}(y, x) : \mathsf{ev}(\lambda x.y(x), x) = y(x)}$$

$$\text{Intro}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\lfloor z, z';\ p : \Pi_{x:A} \mathsf{ev}(z, x) = \mathsf{ev}(z', x) \rfloor\ \mathsf{funext}(z, z', p) : z = z'}$$

$$\text{Prop exp}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\lfloor z, z';\ q : z = z' \rfloor\ \eta_{\mathsf{funext}}(z, z', q) : q = \mathsf{funext}(z, z', \lambda x.\mathsf{ev}(q, x))} \qquad \text{Prop comp}\ \frac{A : \textsc{Type} \quad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\lfloor z, z';\ p : \Pi_{x:A} \mathsf{ev}(z, x) = \mathsf{ev}(z', x) \rfloor\ \beta_{\mathsf{funext}}(z, z', p) : \lambda x.\mathsf{ev}(\mathsf{funext}(z, z', p), x) = p}$$

referring the reader to Figure 5 and for the extended version. We do not explicitly write down the semantics of the propositional extensionality rules, but they can be formulated analogously—we refer the reader to [Spa25] for additional details. Again, we present the semantic counterpart of the propositional expansion rule, which follows automatically from propositional function extensionality—syntactically, one defines:

$$\eta^{\Pi}(z) \equiv \mathsf{funext}(\ \lambda x.\mathsf{ev}(z, x), z, \lambda x.(\ \beta^{\Pi}(\mathsf{ev}(z, -), x)^{-1}\ )\ ) : z = \lambda x.\mathsf{ev}(z, x)$$

in context $z : \Pi_{x:A} B(x)$. We will also refer to the computation and expansion rules as β-reduction and η-expansion, respectively.

Definition 2.14 (Semantic propositional dependent product types)**.** We say that a category with attributes $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **equipped with semantic propositional dependent product types** if it is equipped with semantic propositional identity types, it satisfies *formation*, *introduction*, *elimination*, *compatibility with the substitution* of Definition 2.13 and moreover:

- (*Propositional computation and additional compatibility with the substitution*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic term $\Gamma.A \xrightarrow{b} \Gamma.A.B$ of B and every semantic term $\Gamma \xrightarrow{a} \Gamma.A$ of A, there is a choice of a semantic term:

$$\Gamma \xrightarrow{\beta^a_b} \Gamma.\mathsf{id}_{B[a]}[\ \mathsf{ev}^a_{\lambda b}; b[a]\ ]$$

of type $\mathsf{id}_{B[a]}[\ \mathsf{ev}^a_{\lambda b}; b[a]\ ]$, in such a way that the diagram:

$$
\begin{array}{ccc}
\Delta & \xrightarrow{\quad \beta^a_b[f] \quad} & \Delta.\mathsf{id}_{B[a]}[\ \mathsf{ev}^a_{\lambda b}; b[a]\ ][f] \\
\Vert & & \Vert \ \text{Lemma 2.6} \\
\Vert & & \Delta.\mathsf{id}_{B[a]}[\ f.B[a].B[aP_{B[a]}]\ ][\ \mathsf{ev}^a_{\lambda b}[f]; b[a][f]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{B[a][f]}[\ \mathsf{ev}^a_{\lambda b}[f]; b[a][f]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{B[f.A][a[f]]}[\ \mathsf{ev}^a_{\lambda b}[f]; b[f.A][a[f]]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{B[f.A][a[f]]}[\ \mathsf{ev}^{a[f]}_{(\lambda b)[f]}; b[f.A][a[f]]\ ] \\
\Vert & & \Vert \\
\Delta & \xrightarrow{\quad \beta^{a[f]}_{b[f.A]} \quad} & \Delta.\mathsf{id}_{B[f.A][a[f]]}[\ \mathsf{ev}^{a[f]}_{\lambda(b[f.A])}; b[f.A][a[f]]\ ]
\end{array}
$$

commutes i.e. $\beta^a_b[f] = \beta^{a[f]}_{b[f.A]}$, for every morphism of contexts $\Delta \xrightarrow{f} \Gamma$;

- (*Propositional expansion and additional compatibility with the substitution*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$ and every semantic term $\Gamma \xrightarrow{z} \Gamma.\mathbf{\Pi}^B_A$ of $\mathbf{\Pi}^B_A$, there is a choice of a semantic term:

$$\Gamma \xrightarrow{\eta_z} \Gamma.\mathsf{id}_{\mathbf{\Pi}^B_A}[\ \lambda\mathsf{ev}^{v_A}_{z[P_A]}; z\ ]$$

of type $\mathsf{id}_{\mathbf{\Pi}^B_A}[\ \lambda\mathsf{ev}^{v_A}_{z[P_A]}; z\ ]$, in such a way that the diagram:

$$
\begin{array}{ccc}
\Delta & \xrightarrow{\quad \eta_z[f] \quad} & \Delta.\mathsf{id}_{\mathbf{\Pi}^B_A}[\ \lambda\mathsf{ev}^{v_A}_{z[P_A]}; z\ ][f] \\
\Vert & & \Vert \ \text{Lemma 2.6} \\
\Vert & & \Delta.\mathsf{id}_{\mathbf{\Pi}^B_A}[\ f.\mathbf{\Pi}^B_A.\mathbf{\Pi}^B_A[P_{\mathbf{\Pi}^B_A}]\ ][\ (\lambda\mathsf{ev}^{v_A}_{z[P_A]})[f]; z[f]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{\mathbf{\Pi}^B_A[f]}[\ (\lambda\mathsf{ev}^{v_A}_{z[P_A]})[f]; z[f]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{\mathbf{\Pi}^{B[f.A]}_{A[f]}}[\ \lambda(\mathsf{ev}^{v_A}_{z[P_A]}[f.A]); z[f]\ ] \\
\Vert & & \Vert \\
\Vert & & \Delta.\mathsf{id}_{\mathbf{\Pi}^{B[f.A]}_{A[f]}}[\ \lambda(\mathsf{ev}^{v_A[f.A]}_{z[P_A][f.A]}); z[f]\ ] \\
\Vert & & \Vert \ \text{Lemma 2.8} \\
\Delta & \xrightarrow{\quad \eta_{z[f]} \quad} & \Delta.\mathsf{id}_{\mathbf{\Pi}^{B[f.A]}_{A[f]}}[\ \lambda(\mathsf{ev}^{v_{A[f]}}_{z[f][P_{A[f]}]}); z[f]\ ]
\end{array}
$$

commutes i.e. $\eta_z[f] = \eta_{z[f]}$, for every morphism of contexts $\Delta \xrightarrow{f} \Gamma$.

The following two notions (Definition 2.15 and Definition 2.16) define when a category with attributes is equipped with semantic dependent sum types (in the extensional and propositional case, respectively). In detail, the following notion is meant to model the inference rules for the dependent sum types, that here we recall in a concise form:

$$\text{Form}\ \frac{A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\Sigma_{x:A} B(x) : \textsc{Type}} \qquad \text{Elim}\ \frac{A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \qquad \lfloor u : \Sigma_{x:A} B(x) \rfloor\ C(u) : \textsc{Type} \qquad \lfloor x : A;\ y : B(x) \rfloor\ c(x,y) : C(\langle x, y\rangle)}{\lfloor u : \Sigma_{x:A} B(x) \rfloor\ \mathsf{split}(c, u) : C(u)}$$

$$\text{Intro}\ \frac{A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type}}{\lfloor x : A;\ y : B(x) \rfloor\ \langle x, y \rangle : \Sigma_{x:A} B(x)} \qquad \text{Comp}\ \frac{A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \qquad \lfloor u : \Sigma_{x:A} B(x) \rfloor\ C(u) : \textsc{Type} \qquad \lfloor x : A;\ y : B(x) \rfloor\ c(x,y) : C(\langle x, y\rangle)}{\lfloor x : A;\ y : B(x) \rfloor\ \mathsf{split}(c, \langle x, y \rangle) \equiv c(x,y)}$$

referring the reader to Figure 6 for the extended version.

Definition 2.15 (Semantic dependent sum types)**.** A category with attributes $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **equipped with semantic dependent sum types** if:

- (*Formation and introduction*) for every semantic context Γ, every semantic type A in context Γ and every semantic type B in context $\Gamma.A$, there is a choice of a semantic type $\mathbf{\Sigma}_A^B$ and of a morphism of contexts $\Gamma.A.B \xrightarrow{\mathsf{p}_A^B} \Gamma.\mathbf{\Sigma}_A^B$ such that:

$$\begin{array}{ccc} \Gamma.A.B & \xrightarrow{\mathsf{p}_A^B} & \Gamma.\mathbf{\Sigma}_A^B \\ \big\downarrow{\scriptstyle P_B} & & \big\downarrow{\scriptstyle P_{\mathbf{\Sigma}_A^B}} \\ \Gamma.A & \xrightarrow{P_A} & \Gamma \end{array}$$

  commutes;
- (*Elimination*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic type C in context $\Gamma.\mathbf{\Sigma}_A^B$ and every semantic term c of type $C[\mathsf{p}_A^B]$ in context $\Gamma.A.B$, there is a choice of a semantic term split_c of type C in context $\Gamma.\mathbf{\Sigma}_A^B$;

in such a way that the following properties are satisfied:

- (*Computation*) For every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic type C in context $\Gamma.\mathbf{\Sigma}_A^B$ and every semantic term c of type $C[\mathsf{p}_A^B]$ in context $\Gamma.A.B$, the equality:

$$(\ \Gamma.A.B \xrightarrow{\mathsf{split}_c[\mathsf{p}_A^B]} \Gamma.A.B.C[\mathsf{p}_A^B]\ ) = (\ \Gamma.A.B \xrightarrow{c} \Gamma.A.B.C[\mathsf{p}_A^B]\ )$$

  between semantic terms of type $C[\mathsf{p}_A^B]$ holds.
- (*Compatibility with the substitution*) For every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic type C in context $\Gamma.\mathbf{\Sigma}_A^B$, every semantic term c of type $C[\mathsf{p}_A^B]$ in context $\Gamma.A.B$, if $\Delta \xrightarrow{f} \Gamma$ is a morphism of semantic contexts, then:
  - the equality:

$$\mathbf{\Sigma}_A^B[f] = \mathbf{\Sigma}_{A[f]}^{B[f.A]}$$

    between semantic types in context Γ holds;

– the diagram:

$$\begin{array}{ccccc}
\Delta.A[f].B[f.A] & & \xrightarrow{\quad f.A.B \quad} & & \Gamma.A.B \\
\downarrow {\scriptstyle \mathsf{p}_{A[f]}^{B[f.A]}} & & & & \downarrow {\scriptstyle \mathsf{p}_A^B} \\
\Delta.\boldsymbol{\Sigma}_{A[f]}^{B[f.A]} & = & \Delta.\boldsymbol{\Sigma}_A^B[f] & \xrightarrow{f.\boldsymbol{\Sigma}_A^B} & \Gamma.\boldsymbol{\Sigma}_A^B
\end{array}$$

commutes;

– the equality between the semantic terms:

$$\Delta.\boldsymbol{\Sigma}_A^B[f] \xrightarrow{\mathsf{split}_c[f.\boldsymbol{\Sigma}_A^B]} \Delta.\boldsymbol{\Sigma}_A^B[f].C[f.\boldsymbol{\Sigma}_A^B]$$
$$\Delta.\boldsymbol{\Sigma}_{A[f]}^{B[f.A]} \xrightarrow{\mathsf{split}_{c[f.A.B]}} \Delta.\boldsymbol{\Sigma}_{A[f]}^{B[f.A]}.C[f.\boldsymbol{\Sigma}_A^B]$$

of semantic type $C[f.\boldsymbol{\Sigma}_A^B]$ in semantic context $\Delta.\boldsymbol{\Sigma}_A^B[f] = \Delta.\boldsymbol{\Sigma}_{A[f]}^{B[f.A]}$ holds, where we remind that $c[f.A.B]$ is a term:

$$\Delta.A[f].B[f.A] \ \to \ \Delta.A[f].B[f.A].\big(\, C\big[\mathsf{p}_A^B(f.A.B)\big] \ = \ C\big[f.\boldsymbol{\Sigma}_A^B\big]\big[\mathsf{p}_{A[f]}^{B[f.A]}\big] \,\big).$$

The following notion is meant to model the inference rules for the propositional dependent sum types, that here we recall in a concise form:

$$\text{Form}\ \frac{\begin{array}{c} A : \textsc{Type} \\ \lfloor x : A \rfloor\ B(x) : \textsc{Type} \end{array}}{\Sigma_{x:A} B(x) : \textsc{Type}} \qquad \text{Elim}\ \frac{\begin{array}{c} A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \\ \lfloor u : \Sigma_{x:A} B(x) \rfloor\ C(u) : \textsc{Type} \\ \lfloor x : A;\ y : B(x) \rfloor\ c(x,y) : C(\langle x, y \rangle) \end{array}}{\lfloor u : \Sigma_{x:A} B(x) \rfloor\ \mathsf{split}(c, u) : C(u)}$$

$$\text{Intro}\ \frac{\begin{array}{c} A : \textsc{Type} \\ \lfloor x : A \rfloor\ B(x) : \textsc{Type} \end{array}}{\begin{array}{c} \lfloor x : A;\ y : B(x) \rfloor \\ \langle x, y \rangle : \Sigma_{x:A} B(x) \end{array}} \qquad \text{Prop comp}\ \frac{\begin{array}{c} A : \textsc{Type} \qquad \lfloor x : A \rfloor\ B(x) : \textsc{Type} \\ \lfloor u : \Sigma_{x:A} B(x) \rfloor\ C(u) : \textsc{Type} \\ \lfloor x : A;\ y : B(x) \rfloor\ c(x,y) : C(\langle x, y \rangle) \end{array}}{\begin{array}{c} \lfloor x : A;\ y : B(x) \rfloor \\ \sigma(c, x, y) : \mathsf{split}(c, \langle x, y \rangle) = c(x, y) \end{array}}$$

referring the reader to Figure 7 for the extended version.

Definition 2.16 (Semantic propositional dependent sum types)**.** We say that $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **equipped with semantic propositional dependent sum types** if it is equipped with semantic propositional identity types, it satisfies *formation and introduction*, *elimination*, *compatibility with the substitution* of Definition 2.15 and moreover:

- (*Propositional computation*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic type C in context $\Gamma.\boldsymbol{\Sigma}_A^B$ and every semantic term c of type $C[\mathsf{p}_A^B]$ in context $\Gamma.A.B$, there is a choice of a semantic term:
$$\Gamma.A.B \xrightarrow{\sigma_c} \Gamma.A.B.\mathsf{id}_{C[\mathsf{p}_A^B]}[\ \mathsf{split}_c[\mathsf{p}_A^B]; c\ ]$$
of type $\mathsf{id}_{C[\mathsf{p}_A^B]}[\ \mathsf{split}_c[\mathsf{p}_A^B]; c\ ]$;

- (*Additional compatibility with the substitution*) for every semantic context Γ, every semantic type A in context Γ, every semantic type B in context $\Gamma.A$, every semantic type C in context $\Gamma.\mathbf{\Sigma}_A^B$, every semantic term c of type $C[\mathsf{p}_A^B]$ in context $\Gamma.A.B$ and every morphism of semantic contexts $\Delta \xrightarrow{f} \Gamma$, the following diagram:

$$\begin{array}{ccc}
\Delta.A[f].B[f.A] & \xrightarrow{\sigma_{c[f.A.B]}} & \Delta.A[f].B[f.A].\mathsf{id}_{C[f.\mathbf{\Sigma}_A^B][\mathsf{p}_{A[f]}^{B[f.A]}]}[\ \mathsf{split}_{c[f.A.B]}[\mathsf{p}_{A[f]}^{B[f.A]}]; c[f.A.B]\ ] \\
\Vert & & \Vert \\
 & & \Delta.A[f].B[f.A].\mathsf{id}_{C[\mathsf{p}_A^B][f.A.B]}[\ \mathsf{split}_{c[f.A.B]}[\mathsf{p}_{A[f]}^{B[f.A]}]; c[f.A.B]\ ] \\
 & & \Vert \\
 & & \Delta.A[f].B[f.A].\mathsf{id}_{C[\mathsf{p}_A^B][f.A.B]}[\ \mathsf{split}_c[f.\mathbf{\Sigma}_A^B][\mathsf{p}_{A[f]}^{B[f.A]}]; c[f.A.B]\ ] \\
 & & \Vert \\
 & & \Delta.A[f].B[f.A].\mathsf{id}_{C[\mathsf{p}_A^B][f.A.B]}[\ \mathsf{split}_c[\mathsf{p}_A^B][f.A.B]; c[f.A.B]\ ] \\
 & & \Vert \\
 & & \Delta.A[f].B[f.A].\mathsf{id}_{C[\mathsf{p}_A^B]}[\ f.A.B.C[\mathsf{p}_A^B].C' \\
 & & \Vert \text{ Lemma 2.6} \\
\Delta.A[f].B[f.A] & \xrightarrow[\sigma_c[f.A.B]]{} & \Delta.A[f].B[f.A].\mathsf{id}_{C[\mathsf{p}_A^B]}[\ \mathsf{split}_c[\mathsf{p}_A^B]; c\ ][f.A.B]
\end{array}$$

where $C' \equiv C[\mathsf{p}_A^B P_{C[\mathsf{p}_A^B]}\ ][\ \mathsf{split}_c[\mathsf{p}_A^B][f.A.B]; c[f.A.B]\ ]$, commutes i.e. $\sigma_{c[f.A.B]} = \sigma_c[f.A.B]$.

We have all the notions that we need to give the following:

Definition 2.17. Let $(\mathcal{C}, \textsc{tp}, -.-, P)$ be a category with attributes.

- If T is a given extensional type theory, we say that $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **model of T** if it is equipped with semantic extensional identity types, with semantic dependent product types, with semantic dependent sum types and with a choice of a semantic type (and of a semantic term) in context $\mathtt{1}$ for every atomic type (and every atomic term) of T.
- If T is a given propositional type theory, we say that $(\mathcal{C}, \textsc{tp}, -.-, P)$ is **model of T** if it is equipped with semantic propositional identity types, with semantic propositional dependent product types, with semantic propositional dependent sum types and with a choice of a semantic type (and of a semantic term) in context $\mathtt{1}$ for every atomic type (and every atomic term) of T.

Definition 2.18. Let $(\mathcal{C}, \textsc{tp}_\mathcal{C}, -.-, P_\mathcal{C})$ and $(\mathcal{D}, \textsc{tp}_\mathcal{D}, -.-, P_\mathcal{D})$ be categories with attributes. A **morphism of categories with attributes**:

$$F : (\mathcal{C}, \textsc{tp}_\mathcal{C}, -.-, P_\mathcal{C}) \to (\mathcal{D}, \textsc{tp}_\mathcal{D}, -.-, P_\mathcal{D})$$

consists of:

- a functor $F : \mathcal{C} \to \mathcal{D}$ preserving the terminal object;
- a natural transformation, that we continue calling F, from $\textsc{tp}_\mathcal{C}$ to $\textsc{tp}_\mathcal{D} F$, i.e. the equality:

$$F(A[f]) = (FA)[Ff]$$

holds for every semantic type A in some semantic context Γ of $\mathcal{C}$ and every morphism of semantic contexts $\Delta \xrightarrow{f} \Gamma$ of $\mathcal{C}$;

in such a way that the equality:

$$F(-.-) = (F-).(F-)$$

holds and that $FP_A = P_{FA}$ for every semantic type A in some semantic context Γ of $\mathcal{C}$.

Moreover, if $(\mathcal{C}, \text{TP}_\mathcal{C}, -.-, P_\mathcal{C})$ and $(\mathcal{D}, \text{TP}_\mathcal{D}, -.-, P_\mathcal{D})$ are models of a given extensional (propositional, respectively) type theory T, then a **morphism of T**:

$$(\mathcal{C}, \text{TP}_\mathcal{C}, -.-, P_\mathcal{C}) \to (\mathcal{D}, \text{TP}_\mathcal{D}, -.-, P_\mathcal{D})$$

is a morphism of categories with attributes $(\mathcal{C}, \text{TP}_\mathcal{C}, -.-, P_\mathcal{C}) \to (\mathcal{D}, \text{TP}_\mathcal{D}, -.-, P_\mathcal{D})$ preserving the semantic extensional (propositional, respectively) identity types, the semantic (propositional, respectively) dependent product types, the semantic (propositional, respectively) dependent sum types and the choice of the semantic types (and of the semantic term) in context $\mathtt{1}$.

Remark 2.19. The one that we presented in Definition 2.18 happens to be the *strict* notion of morphism between categories with attributes, and it is considered e.g. by Cartmell [Car78] and by Kapulkin and Lumsdaine [KL21]. In this paper we only deal with morphisms between categories with attributes in this strict form: they all strictly commute with the given semantic context extensions and the display maps. The syntactic model of an extensional (propositional) type theory (see Remark 2.21) enjoys a strict universal initiality property with respect to morphisms of the given existential (propositional, respectively) type theory in this strong form.

However several weakenings of this notion are available in the literature, depending on their strictness in the commutativity with the category with attributes structure. E.g. the notion used by Clairambault and Dybjer [CD14] preserves the semantic context extension only up to natural isomorphism. We refer the reader to [New18] for more details (regarding in this case the related structure of natural model).

The propositional type theory hPTT**.** Now, let us consider the given PTT (see Subsection 1.1). Let us consider the propositional type theory contained in PTT (meaning that all of its contexts, types, terms and judgements are contexts, types, terms and judgements of PTT) whose atomic types are the ones of PTT that are provably h-sets in PTT (and whose atomic terms are the atomic terms of these atomic h-sets in PTT). We indicate this specific propositional type theory as hPTT, since the contexts of hPTT are the ones that we call *homotopy elementary* contexts of PTT (see Definition 5.2).

Remark 2.20. Since ETT and hPTT have the same atomic types and terms (see Subsection 1.1), every model of ETT is canonically a model of hPTT: in order to obtain a choice of the terms J_c, H_c, β^a_b, η_z and σ_c, one defines them as instances of r^t_T for opportune semantic types T and semantic terms t. Then, all of the additional compatibilities with the substitution follow in fact by the one of r_T.

With this choice of a structure of model of hPTT for every model of ETT, every morphism of ETT is canonically a morphism of hPTT.

Remark 2.21. The category whose objects are the contexts of ETT and whose arrows are the morphisms of contexts (identified up to renaming their free variables and up to componentwise equality judgement) has for terminal object the empty context and constitutes a category with attributes with the following data:

- the semantic types in a given semantic context $\gamma : \Gamma$ are the type judgements $\lfloor \gamma : \Gamma \rfloor_{\mathtt{ext}}\ A(\gamma) : \text{Type}$ of ETT in context γ; the presheaf of semantic types act on the morphisms of contexts by substitution;

- the semantic context extension maps a pair $(\gamma : \Gamma, \lfloor\gamma\rfloor_{\mathtt{ext}}\ A(\gamma) : \text{Type})$ to the context:

$$\gamma : \Gamma, x : A(\gamma)$$

  and an arrow $\lfloor\delta : \Delta\rfloor_{\mathtt{ext}}\ f(\delta) : \Gamma$ of target $(\gamma : \Gamma, \lfloor\gamma\rfloor_{\mathtt{ext}}\ A(\gamma) : \text{Type})$ to the morphism of contexts:

$$\delta, x' \xrightarrow{\lfloor\delta:\Delta, x':A(f(\delta))\rfloor_{\mathtt{ext}}\ f(\delta):\Gamma, x':A(f(\delta))} \gamma, x;$$

- whenever $\lfloor\gamma\rfloor_{\mathtt{ext}}\ A(\gamma) : \text{Type}$ is a semantic type in semantic context γ i.e. a type judgement in context γ, then the display map $P_{\lfloor\gamma\rfloor_{\mathtt{ext}}\ T(\gamma):\text{Type}}$ is the morphism of contexts:

$$\lfloor\gamma, x : A(\gamma)\rfloor_{\mathtt{ext}}\ \gamma;$$

  its sections—i.e. the semantic terms of $\lfloor\gamma\rfloor_{\mathtt{ext}}\ A(\gamma) : \text{Type}$—are the morphisms of contexts $\gamma \to \gamma, x$ of the form:

$$\lfloor\gamma\rfloor_{\mathtt{ext}}\ \gamma, a(\gamma) : A(\gamma)$$

  for some term judgement $\lfloor\gamma\rfloor_{\mathtt{ext}}\ a(\gamma) : A(\gamma)$.

We indicate as **eTT** this category with attributes. Then **eTT** is a model of ETT with the clear choices of the semantic extensional identity types, of the semantic dependent sum types, and of the semantic dependent product types. The choice of a semantic type (and of a semantic term) in empty context for every atomic type (and every atomic term) is the identity.

Analogously, PTT and hPTT form models of themselves, which we indicate as **pTT** and **hpTT**.

Theorem 2.22 (Soundness)**.** *For every model* $(\mathcal{C}, \text{TP}, -.-, P)$ *of* ETT*, there is unique a morphism* **eTT** $\to (\mathcal{C}, \text{TP}, -.-, P)$ *of* ETT*.*

Analogously, for every model $(\mathcal{C}, \text{TP}, -.-, P)$ *of* PTT*, there is unique a morphism:*

$$\mathbf{pTT} \to (\mathcal{C}, \text{TP}, -.-, P)$$

of PTT *and, for every model* $(\mathcal{C}, \text{TP}, -.-, P)$ *of* hPTT*, there is unique a morphism:*

$$\mathbf{hpTT} \to (\mathcal{C}, \text{TP}, -.-, P)$$

of hPTT*.*

We conclude this subsection by noticing the following Remark 2.23. For further details on this general notion of semantics (or equivalent ones) for dependent type theories, we refer the reader to [Car78, Pit00, Str91].

Remark 2.23. By Remark 2.20 and by Theorem 2.22, there is unique a morphism:

$$\mathbf{hpTT} \to \mathbf{eTT}$$

of hPTT. We call it **canonical interpretation** of hPTT into ETT and denote it as $|\cdot|$. A priori, being a morphism of hPTT, the mapping $|\cdot|$ is defined on contexts, morphisms of contexts and type judgements: we now show how to extend the mapping $|\cdot|$ to types and terms in context.

Let $\gamma : \Gamma$ and let $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ be a type judgement of hPTT in context γ (i.e. an *h-elementary* type judgement of PTT in *h-elementary* context γ—see Definition 5.1 and Definition 5.2). As $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ is a semantic type in semantic context γ in **hpTT**, then $|\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}|$ needs to be a semantic type in semantic context $|\gamma|$ in **eTT** i.e. a type judgement of ETT in context $|\gamma|$. Hence $|\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}|$ is of the form:

$$\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |A(\gamma)| : \text{Type}$$

where $|A(\gamma)|$ denotes therefore a type of ETT in context $|\gamma|$.

Let $\lfloor\gamma\rfloor\ a(\gamma) : A(\gamma)$ be a term judgement of hPTT. Then the morphism $\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)$ is a section of $P_{\lfloor\gamma\rfloor\ A(\gamma):\text{Type}}$ in $\mathtt{hp}$**TT** and therefore $|\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)|$ is a section $|\gamma| \to |\gamma|, |x| : |A(\gamma)|$ of:

$$|P_{\lfloor\gamma\rfloor\ A(\gamma):\text{Type}}| = P_{|\lfloor\gamma\rfloor\ A(\gamma):\text{Type}|} = P_{\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |A(\gamma)|:\text{Type}} = \lfloor|\gamma|, |x| : |A(\gamma)|\rfloor_{\mathtt{ext}}\ |\gamma|$$

in **eTT**. Hence $|\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)|$ needs to be of the form:

$$\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |\gamma|, |a(\gamma)| : |A(\gamma)|$$

for a term judgement $\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |a(\gamma)| : |A(\gamma)|$, where $|a(\gamma)|$ denotes therefore a term of ETT in context $|\gamma|$ and of type $|A(\gamma)|$.

If $\lfloor\delta\rfloor\ f(\delta) : \Gamma$ is a morphism of contexts of $\mathtt{hp}$**TT** and if we write $|A|(|\gamma|)$ and $|a|(|\gamma|)$ for $|A(\gamma)|$ and $|a(\gamma)|$—respectively—in context $|\gamma|$, we observe that:

$$\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |A(f(\delta))| \equiv |A|(|f(\delta)|) \quad \text{and} \quad \lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |a(f(\delta))| = |a|(|f(\delta)|)$$

because $|\cdot|$ needs to commute with the substitution.

3. Homotopy equivalences of contexts in Propositional Type Theory

This section mostly deals with the notion of morphism of contexts and the concept of generalised identity type of two parallel morphisms of context. Here we briefly recap these notions as long as we need them. For more details, we refer the reader to [GG08].

Suppose that we are given two contexts $\gamma : \Gamma$ and $\delta : \Delta$ of PTT, where the former is an abbreviation for the list $\gamma_1 : \Gamma_1, \gamma_2 : \Gamma_2(\gamma_1), ..., \gamma_n : \Gamma_n(\gamma_1, ..., \gamma_{n-1})$ and the latter for $\delta_1 : \Delta_1, \delta_2 : \Delta_2(\delta_1), ..., \delta_m : \Delta_m(\delta_1, ..., \delta_{m-1})$. With the expression $\lfloor\gamma : \Gamma\rfloor\ a(\gamma) : \Delta$ we mean a list of judgements:

- $\lfloor\gamma : \Gamma\rfloor\ a_1(\gamma) : \Delta_1$
- $\lfloor\gamma : \Gamma\rfloor\ a_2(\gamma) : \Delta_2(a_1(\gamma))$
- $\lfloor\gamma : \Gamma\rfloor\ a_3(\gamma) : \Delta_3(a_1(\gamma), a_2(\gamma))$
- ...
- $\lfloor\gamma : \Gamma\rfloor\ a_m(\gamma) : \Delta_m(a_1(\gamma), a_2(\gamma), a_3(\gamma), ..., a_{m-1}(\gamma))$

and we call such a list a *morphism of contexts* $\gamma \to \delta$. Now, suppose that $\delta' : \Delta'$ indicates the context:

$$\delta_1 : \Delta_1, ..., \delta_{m-1} : \Delta_{m-1}(\delta_1, ..., \delta_{m-2})$$

and $\lfloor\gamma : \Gamma\rfloor\ a'(\gamma) : \Delta'$ indicates the list:

- $\lfloor\gamma : \Gamma\rfloor\ a_1(\gamma) : \Delta_1$
- $\lfloor\gamma : \Gamma\rfloor\ a_2(\gamma) : \Delta_2(a_1(\gamma))$
- $\lfloor\gamma : \Gamma\rfloor\ a_3(\gamma) : \Delta_3(a_1(\gamma), a_2(\gamma))$
- ...
- $\lfloor\gamma : \Gamma\rfloor\ a_{m-1}(\gamma) : \Delta_m(a_1(\gamma), a_2(\gamma), a_3(\gamma), ..., a_{m-2}(\gamma))$

and $A(\delta')$ indicates the type in context $\Delta_m(\delta_1, ..., \delta_{m-1})$. Then we may also write anyone of the following:

$$\lfloor\gamma : \Gamma\rfloor\ a(\gamma) : \Delta', A \qquad \lfloor\gamma : \Gamma\rfloor\ a'(\gamma), a_m(\gamma) : \Delta$$

$$\lfloor\gamma : \Gamma\rfloor\ a'(\gamma), a_m(\gamma) : \Delta', A \qquad \lfloor\gamma : \Gamma\rfloor\ a'(\gamma) : \Delta', a_m(\gamma) : A(a'(\gamma))$$

in order to indicate the same morphism of contexts $\lfloor\gamma : \Gamma\rfloor\ a(\gamma) : \Delta$.

If we are given two parallel morphisms of contexts $\lfloor\gamma\rfloor\ a(\gamma) : \Delta$ and $\lfloor\gamma\rfloor\ b(\gamma) : \Delta$, the expression $\lfloor\gamma\rfloor\ p(\gamma) : a(\gamma) = b(\gamma)$ indicates the list:

- $\lfloor\gamma\rfloor\ p_1(\gamma) : a_1(\gamma) = b_1(\gamma)$
- $\lfloor\gamma\rfloor\ p_2(\gamma) : a_2(\gamma) = p_1(\gamma)^* b_2(\gamma)$
- $\lfloor\gamma\rfloor\ p_3(\gamma) : a_3(\gamma) = (p_1(\gamma), p_2(\gamma))^* b_3(\gamma)$
- ...
- $\lfloor\gamma\rfloor\ p_m(\gamma) : a_m(\gamma) = (p_1(\gamma), ..., p_{m-1}(\gamma))^* b_m(\gamma)$

where the operations $(p_1(\gamma), ..., p_k(\gamma))^*$ are defined by sequential (generalised) path inductions on p_k, p_{k-1}, ..., p_2, and p_1 and hence make the identity types:

$$(\ \mathsf{r}(a_1(\gamma)), ..., \mathsf{r}(a_k(\gamma))\ )^* d = d$$

inhabited (remind that PTT has *propositional* identity types) if $d : \Delta_{k+1}(b_1(\gamma), ..., b_k(\gamma))$. We call *context propositional equality* this new meaning of the symbol $=$ in between two parallel morphisms of contexts.

As shown by Gambino and Garner [GG08, Gar09], the expression $\lfloor\gamma : \Gamma\rfloor\ p(\gamma) : a(\gamma) = b(\gamma)$ formally verifies the same rules of Figure 3 verified by the propositional equality and one can simply prove this by sequential (generalised) path induction. In particular, a context elimination (i.e. path induction) rule, with a corresponding context propositional computation rule, is satisfied by the context propositional equality, and moreover every context homotopy equivalence—see Definition 3.1—is also a context half-adjoint equivalence.

Definition 3.1. A **context homotopy equivalence** between $\gamma : \Gamma$ and $\delta : \Delta$ is a couple of morphisms of contexts of the form:

- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}(\gamma) : \Delta$
- $\lfloor\delta : \Delta\rfloor\ \mathtt{g}(\delta) : \Gamma$

such that *there exist* couples of judgements of the form:

- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}(\gamma) : \mathtt{g}(\mathtt{f}(\gamma)) = \gamma$
- $\lfloor\delta : \Delta\rfloor\ \mathtt{q}(\delta) : \mathtt{f}(\mathtt{g}(\delta)) = \delta$

hence, as usual, the expression $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}(\gamma) : \Delta$ is an abbreviation for the list of judgements:

- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}_1(\gamma) : \Delta_1$
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}_2(\gamma) : \Delta_2(\mathtt{f}_1(\gamma))$
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}_3(\gamma) : \Delta_3(\mathtt{f}_1(\gamma), \mathtt{f}_2(\gamma))$
- ...
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{f}_m(\gamma) : \Delta_m(\mathtt{f}_1(\gamma), \mathtt{f}_2(\gamma), \mathtt{f}_3(\gamma), ..., \mathtt{f}_{m-1}(\gamma))$

and the expression $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}(\gamma) : \mathtt{g}(\mathtt{f}(\gamma)) = \gamma$ is an abbreviation for the list of judgements:

- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}_1(\gamma) : \mathtt{g}_1(\mathtt{f}(\gamma)) = \gamma_1$
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}_2(\gamma) : \mathtt{g}_2(\mathtt{f}(\gamma)) = \mathtt{p}_1(\gamma)^* \gamma_2$
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}_3(\gamma) : \mathtt{g}_3(\mathtt{f}(\gamma)) = (\mathtt{p}_1(\gamma), \mathtt{p}_2(\gamma))^* \gamma_3$
- ...
- $\lfloor\gamma : \Gamma\rfloor\ \mathtt{p}_n(\gamma) : \mathtt{g}_n(\mathtt{f}(\gamma)) = (\mathtt{p}_1(\gamma), \mathtt{p}_2(\gamma), ..., \mathtt{p}_{n-1}(\gamma))^* \gamma_n$.

In Subsection 3.1 we show under what hypotheses these equivalences can be extended to wider contexts.

3.1. **Extension of context homotopy equivalences.** In what follows, and throughout the entire paper, we will often adopt the following conventions for naming variables: if we denote two given types by A and A', and denote a variable of type A by x, then we will often write $\underline{x}$ to denote a variable of type A'. Additionally, we will use x', x'', etc. (and similarly $\underline{x}'$, $\underline{x}''$, etc.) to denote variables typed by re-indexings of A (and of A', respectively).

Let $\gamma : \Gamma$ and $\delta : \Delta$ be contexts and let us assume that we are given a context homotopy equivalence $(\mathtt{f};\mathtt{g})$ as follows:

- $\lfloor \gamma : \Gamma \rfloor\ \mathtt{f}(\gamma) : \Delta$
- $\lfloor \delta : \Delta \rfloor\ \mathtt{g}(\delta) : \Gamma$

between them as before. Suppose that we are given the judgements:

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$
- $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f(\gamma, x) : A'(\mathtt{f}(\gamma))$
- $\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma, \underline{x}') : A(\gamma)$

such that there are terms of the form:

- $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ p(\gamma, x) : x = g(\gamma, f(\gamma, x))$
- $\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ q(\gamma, \underline{x}') : \underline{x}' = f(\gamma, g(\gamma, \underline{x}'))$.

In other words, we are given a *homotopy equivalence* $(f;g)$ *between the types in context* $A(\gamma)$ *and* $A'(\delta)$ *relative to the context homotopy equivalence* $(\mathtt{f};\mathtt{g})$. We can use these data in order to augment the context homotopy equivalence $(\mathtt{f};\mathtt{g})$ to a context homotopy equivalence between the contexts:

$$\gamma : \Gamma, x : A(\gamma) \text{ and } \delta : \Delta, \underline{x} : A'(\delta).$$

The remainder of the current subsection is devoted to showing this construction: *the reader who is willing to skip the details may turn to Lemma 3.2.*

We observe that $\lfloor \delta : \Delta, \underline{x}'' : A'(\mathtt{f}(\mathtt{g}(\delta))) \rfloor\ g(\mathtt{g}(\delta), \underline{x}'') : A(\mathtt{g}(\delta))$ and that:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{q}(\delta)^*\underline{x} : A'(\mathtt{f}(\mathtt{g}(\delta)))$$

hence $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ g(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x}) : A(g(\delta))$. Let us rename:

- $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ \mathtt{f}_{m+1}(\gamma, x) \equiv f(\gamma, x) : A'(\mathtt{f}(\gamma))$
- $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{g}_{n+1}(\delta, \underline{x}) \equiv g(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x}) : A(\mathtt{g}(\delta))$.

We observe that:

$$\mathtt{f}_{m+1}(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x})) \equiv f(\mathtt{g}(\delta), g(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x})) \overset{q(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x})}{=} \mathtt{q}(\delta)^*\underline{x}$$

in context $\delta : \Delta, \underline{x} : A'(\delta)$. Moreover:

$$\begin{aligned}\mathtt{g}_{n+1}(\mathtt{f}(\gamma), \mathtt{f}_{m+1}(\gamma, x)) &\equiv g(\mathtt{g}(\mathtt{f}(\gamma)), \mathtt{q}(\mathtt{f}(\gamma))^*\mathtt{f}_{m+1}(\gamma, x)) \\ &= g(\mathtt{g}(\mathtt{f}(\gamma)), \mathtt{f}(\mathtt{p}(\gamma))^*\mathtt{f}_{m+1}(\gamma, x)) \\ &= \mathtt{p}(\gamma)^* x\end{aligned}$$

in context $\gamma : \Gamma, x : A(\gamma)$, where the first identity type is inhabited as one can assume w.l.o.g. that:

$$\lfloor \gamma : \Gamma \rfloor\ \mathtt{q}(\mathtt{f}(\gamma)) = \mathtt{f}(\mathtt{p}(\gamma))$$

and the second by applying based (generalised) path induction n times on $\mathtt{p}_1(\gamma)$, $\mathtt{p}_2(\gamma)$, ..., $\mathtt{p}_n(\gamma)$ and finally using that $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ p(\gamma, x) : g(\gamma, f(\gamma, x)) = x$.

In conclusion, we saw that we can augment the contexts $\gamma : \Gamma$ and $\delta : \Delta$ in:

$$\gamma : \Gamma, x : A(\gamma) \text{ and } \delta : \Delta, \underline{x} : A'(\delta)$$

respectively, in such a way that they continue being homotopy equivalent. In fact the extended context morphisms:

- $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ \mathtt{f}(\gamma) : \Delta, \mathtt{f}_{m+1}(\gamma, x) : A'(\mathtt{f}(\gamma))$ i.e. $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ (\mathtt{f}, \mathtt{f}_{m+1})(\gamma, x) : \Delta, A'$
- $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{g}(\delta) : \Gamma, \mathtt{g}_{n+1}(\delta, \underline{x}) : A(\mathtt{g}(\delta))$ i.e. $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})(\delta, \underline{x}) : \Gamma, A$

constitute a homotopy equivalence, since:

$$\lfloor \gamma : \Gamma \rfloor\ \mathtt{p}(\gamma) : \mathtt{g}(\mathtt{f}(\gamma)) = \gamma \quad \text{and} \quad \lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ \mathtt{g}_{n+1}(\mathtt{f}(\gamma), \mathtt{f}_{m+1}(\gamma, x)) = \mathtt{p}(\gamma)^* x$$

that is:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ \mathtt{g}(\mathtt{f}(\gamma)), \mathtt{g}_{n+1}(\mathtt{f}(\gamma), \mathtt{f}_{m+1}(\gamma, x)) = \gamma, x \quad \text{i.e.}$$
$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})((\mathtt{f}, \mathtt{f}_{m+1})(\gamma, x)) = \gamma, x$$

and since:

$$\lfloor \delta : \Delta \rfloor\ \mathtt{q}(\delta) : \mathtt{f}(\mathtt{g}(\delta)) = \delta \quad \text{and} \quad \lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{f}_{m+1}(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x})) = \mathtt{q}(\delta)^* \underline{x}$$

that is:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{f}(\mathtt{g}(\delta)), \mathtt{f}_{m+1}(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x})) = \delta, \underline{x} \quad \text{i.e.}$$
$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ (\mathtt{f}, \mathtt{f}_{m+1})((\mathtt{g}, \mathtt{g}_{n+1})(\delta, \underline{x})) = \delta, \underline{x}.$$

Let us summarise this into the following:

Lemma 3.2 (Extension)**.** *Let $\gamma : \Gamma$ and $\delta : \Delta$ and let:*

$$\lfloor \gamma : \Gamma \rfloor\ \mathtt{f}(\gamma) : \Delta$$
$$\lfloor \delta : \Delta \rfloor\ \mathtt{g}(\delta) : \Gamma$$

be a context homotopy equivalence $(\mathtt{f}; \mathtt{g})$*. If we are given types:*

$$\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \text{ and } \lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$$

together with a homotopy equivalence:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ *then:*

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ (\mathtt{f}, \mathtt{f}_{m+1})(\gamma, x) : \Delta, A'$$
$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})(\delta, \underline{x}) : \Gamma, A$$

is a context homotopy equivalence $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$*, where:*

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ \mathtt{f}_{m+1}(\gamma, x) \equiv f(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \mathtt{g}_{n+1}(\delta, \underline{x}) \equiv g(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x}) : A(\mathtt{g}(\delta)).$$

We call $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ *the* extension of $(\mathtt{f}; \mathtt{g})$ via $(f; g)$*.*

In the next subsections we analyse specific shapes of homotopy equivalences between types relative to a given context homotopy equivalence. The first regards the ones coming from an application of the dependent product constructor.

3.2. **Dependent product of homotopy equivalences.** We start the current subsection by briefly describing the following data:

Data. Let us assume that $\gamma : \Gamma$ and $\delta : \Delta$ are contexts and that $(\mathtt{f};\mathtt{g})$ is a context homotopy equivalence between them. Moreover, let us assume that we are given judgements:

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$

together with a homotopy equivalence $(f_1; g_1)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$, and a homotopy equivalence $(f_2; g_2)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1; g_1)$ (see Lemma 3.2).

Fact. We can use these data to construct a homotopy equivalence $(f^{\Pi}; g^{\Pi})$ between the types $\Pi_{x:A(\gamma)} B(\gamma, x)$ and $\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ relative to $(\mathtt{f};\mathtt{g})$ as follows, leading to Lemma 3.3 below. The reader may go through the proof under the additional assumption that the contexts γ and δ are empty. However, here we present the proof in full generality because there are several (homotopic) ways to define the pair $(f^{\Pi}; g^{\Pi})$ and the one we choose here will determine the style of the subsequent proofs. A similar remark applies to Subsection 3.3 and Subsection 3.4.

Let us start by considering homotopies:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ p_1(\gamma, x) : x = g_1(\gamma, f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ q_1(\gamma, \underline{x}') : \underline{x}' = f_1(\gamma, g_1(\gamma, \underline{x}'))$$

and homotopies:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ p_2(\gamma, x, y) : y = g_2(\gamma, x, f_2(\gamma, x, y))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ q_2(\gamma, x, \underline{y}') : \underline{y}' = f_2(\gamma, x, g_2(\gamma, x, \underline{y}'))$$

as in our assumptions.

Let us fix the context $\gamma : \Gamma, z : \Pi_{x:A(\gamma)} B(\gamma, x)$ and let us observe that $\lfloor \gamma, z, x : A \rfloor\ \mathsf{ev}(z, x) : B(\gamma, x)$, hence:

$$\lfloor \gamma, z, x \rfloor\ f_2(\gamma, x, \mathsf{ev}(z, x)) : B'(\mathtt{f}(\gamma), f_1(\gamma, x)).$$

Since $\lfloor \gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$, then:

$$\lfloor \gamma, z, \underline{x}' \rfloor\ f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}'))) : B'(\mathtt{f}(\gamma), f_1(\gamma, g_1(\gamma, \underline{x}')))$$

hence:

$$\lfloor \gamma, z, \underline{x}' \rfloor\ q_1(\gamma, \underline{x}')^* f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}'))) : B'(\mathtt{f}(\gamma), \underline{x}').$$

We conclude that:

$$\lfloor \gamma, z \rfloor\ f^{\Pi}(\gamma, z) \equiv \lambda \underline{x}' : A'(\mathtt{f}(\gamma))\ .\ q_1(\gamma, \underline{x}')^* f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}')))$$

is a term of type:

$$\Pi_{\underline{x}':A'(\mathtt{f}(\gamma))} B(\mathtt{f}(\gamma), \underline{x}') \equiv [\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma)).$$

Vice versa, let us fix the context $\gamma : \Gamma, \underline{z}' : \Pi_{\underline{x}':A'(\mathtt{f}(\gamma))} B(\mathtt{f}(\gamma), \underline{x}')$ and let us observe that:

$$\lfloor \gamma, \underline{z}', \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor \ \mathsf{ev}(\underline{z}', \underline{x}') : B'(\mathtt{f}(\gamma), \underline{x}')$$

hence $\lfloor \gamma, \underline{z}', x : A(\gamma) \rfloor \ \mathsf{ev}(\underline{z}', f_1(\gamma, x)) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$ and therefore:

$$\lfloor \gamma, \underline{z}', x : A(\gamma) \rfloor \ g_2(\gamma, x, \mathsf{ev}(\underline{z}', f_1(\gamma, x))) : B(\gamma, x).$$

We conclude that:

$$\lfloor \gamma, \underline{z}' \rfloor \ g^{\Pi}(\gamma, \underline{z}') \equiv \lambda x : A(\gamma) \ . \ g_2(\gamma, x, \mathsf{ev}(\underline{z}', f_1(\gamma, x))) : \Pi_{x:A(\gamma)} B(\gamma, x).$$

We claim that $(f^{\Pi}; g^{\Pi})$ is a homotopy equivalence between:

$$\Pi_{x:A(\gamma)} B(\gamma, x) \text{ and } \Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$$

relative to $(\mathtt{f}; \mathtt{g})$. We start by verifying that $g^{\Pi}(\gamma, f^{\Pi}(\gamma, z)) \equiv z$ in context γ, z. Let $\underline{z}' \equiv f^{\Pi}(\gamma, z)$ and let us observe that:

$$\begin{aligned} \mathsf{ev}(\underline{z}', f_1(\gamma, x)) &= q_1(\gamma, f_1(\gamma, x))^* f_2(\gamma, g_1(\gamma, f_1(\gamma, x)), \mathsf{ev}(z, g_1(\gamma, f_1(\gamma, x)))) \\ &= f_1(\gamma, p_1(\gamma, x))^* f_2(\gamma, g_1(\gamma, f_1(\gamma, x)), \mathsf{ev}(z, g_1(\gamma, f_1(\gamma, x)))) \\ &= f_2(\gamma, x, \mathsf{ev}(z, x)) \end{aligned}$$

where the first equality follows by β-reduction, the second because without loss of generality $q_1(\gamma, f_1(\gamma, x)) = f_1(\gamma, p_1(\gamma, x))$ and the third by based (generalised) path induction on $p_1(\gamma, x)$. By propositional functoriality:

$$g_2(\gamma, x, \mathsf{ev}(\underline{z}', f_1(\gamma, x))) = g_2(\gamma, x, f_2(\gamma, x, \mathsf{ev}(z, x))) \overset{p_2(\gamma, x, \mathsf{ev}(z,x))}{=} \mathsf{ev}(z, x)$$

hence by propositional function extensionality:

$$g^{\Pi}(\gamma, f^{\Pi}(\gamma, z)) \equiv g^{\Pi}(\gamma, \underline{z}') = \lambda x : A(\gamma) \ . \ \mathsf{ev}(z, x) = z$$

where the last identity type is inhabited by η-expansion.

Vice versa, let us verify that $f^{\Pi}(\gamma, z) = \underline{z}'$, where $z \equiv g^{\Pi}(\gamma, \underline{z}')$. At first, we observe that:

$$\mathsf{ev}(z, g_1(\gamma, \underline{x}')) = g_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(\underline{z}', f_1(\gamma, g_1(\gamma, \underline{x}'))))$$

hence:

$$\begin{aligned} f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}'))) &= f_2(\gamma, g_1(\gamma, \underline{x}'), g_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(\underline{z}', f_1(\gamma, g_1(\gamma, \underline{x}'))))) \\ &= \mathsf{ev}(\underline{z}', f_1(\gamma, g_1(\gamma, \underline{x}'))) \end{aligned}$$

where the first identity type is inhabited by propositional functoriality and the second one by the term:

$$q_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(\underline{z}', f_1(\gamma, g_1(\gamma, \underline{x}')))).$$

Secondly, we observe that:

$$q_1(\gamma, \underline{x}')^* f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}'))) = q_1(\gamma, \underline{x}')^* \mathsf{ev}(\underline{z}', f_1(\gamma, g_1(\gamma, \underline{x}'))) = \mathsf{ev}(\underline{z}', \underline{x}')$$

where the former equality follows by propositional functoriality and the latter by based (generalised) path induction on $q_1(\gamma, \underline{x}')$. Therefore:

$$f^{\Pi}(\gamma, g^{\Pi}(\gamma, \underline{z}')) \equiv f^{\Pi}(\gamma, z) = \lambda \underline{x}' : A'(\mathtt{f}(\gamma)) \ . \ \mathsf{ev}(\underline{z}', \underline{x}') = \underline{z}'$$

where the first identity type is inhabited by propositional function extensionality and the second by η-expansion.

Let us summarise this into the following:

Lemma 3.3. *Let us assume that* $\gamma : \Gamma$ *and* $\delta : \Delta$ *are contexts and that* $(\mathtt{f};\mathtt{g})$ *is a context homotopy equivalence between them. Moreover, let us assume that we are given judgements:*

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ *and* $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ *and* $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$

together with a homotopy equivalence $(f_1; g_1)$*:*

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ *and* $A'(\delta)$ *relative to* $(\mathtt{f};\mathtt{g})$*, and a homotopy equivalence* $(f_2; g_2)$*:*

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ *and* $B'(\delta, \underline{x})$ *relative to the extension* $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ *of* $(\mathtt{f};\mathtt{g})$ *via* $(f_1; g_1)$ *(see Lemma 3.2).*

In context:

$$\gamma : \Gamma \text{ and } z : \Pi_{x:A(\gamma)} B(\gamma, x) \text{ and } \underline{z}' : [\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma))$$

if we name:

$$\lfloor \gamma, z \rfloor\ f^{\Pi}(\gamma, z) \equiv \lambda \underline{x}' : A'(\mathtt{f}(\gamma))\ .\ q_1(\gamma, \underline{x}')^* f_2(\gamma, g_1(\gamma, \underline{x}'), \mathsf{ev}(z, g_1(\gamma, \underline{x}'))) : [\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma))$$

$$\lfloor \gamma, \underline{z}' \rfloor\ g^{\Pi}(\gamma, \underline{z}') \equiv \lambda x : A(\gamma)\ .\ g_2(\gamma, x, \mathsf{ev}(\underline{z}', f_1(\gamma, x))) : \Pi_{x:A(\gamma)} B(\gamma, x)$$

then $(f^{\Pi}; g^{\Pi})$ *is a homotopy equivalence between* $\Pi_{x:A(\gamma)} B(\gamma, x)$ *and* $\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ *relative to* $(\mathtt{f};\mathtt{g})$*.*

3.3. **Dependent sum of homotopy equivalences.**

As for Subsection 3.2, let us describe some:

Data. Let us assume that we are given a context homotopy equivalence $(\mathtt{f};\mathtt{g})$ between $\gamma : \Gamma$ and $\delta : \Delta$ and judgements:

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$

together with a homotopy equivalence $(f_1; g_1)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$, and a homotopy equivalence $(f_2; g_2)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between the types $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2).

Fact. Again, we can use these data to construct a homotopy equivalence:

$$(f^{\Sigma}; g^{\Sigma})$$

between $\Sigma_{x:A(\gamma)} B(\gamma, x)$ and $\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ relative to $(\mathtt{f};\mathtt{g})$ as follows, leading to Lemma 3.4 below.

As in Subsection 3.2, let us consider homotopies:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor \; p_1(\gamma, x) : g_1(\gamma, f_1(\gamma, x)) = x$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor \; q_1(\gamma, \underline{x}') : f_1(\gamma, g_1(\gamma, \underline{x}')) = \underline{x}'$$

and homotopies:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor \; p_2(\gamma, x, y) : g_2(\gamma, x, f_2(\gamma, x, y)) = y$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor \; q_2(\gamma, x, \underline{y}') : f_2(\gamma, x, g_2(\gamma, x, \underline{y}')) = \underline{y}'$$

as in our assumptions.

Let us fix the context $\gamma : \Gamma, u : \Sigma_{x:A(\gamma)} B(\gamma, x)$ and let us observe that $\pi_1 u : A(\gamma)$ and that $\pi_2 u : B(\gamma, \pi_1 u)$. Therefore $f_1(\gamma, \pi_1 u) : A'(\mathtt{f}(\gamma))$ and $f_2(\gamma, \pi_1 u, \pi_2 u) : B'(\mathtt{f}(\gamma), f_1(\gamma, \pi_1 u))$ hence:

$$\lfloor \gamma, u \rfloor \; f^{\Sigma}(\gamma, u) \equiv \langle f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u) \rangle : [\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma)).$$

Vice versa, if $\gamma : \Gamma, \underline{u}' : \Sigma_{\underline{x}':A'(\mathtt{f}(\gamma))} B'(\mathtt{f}(\gamma), \underline{x}')$ then $\pi_1 \underline{u}' : A'(\mathtt{f}(\gamma))$ hence $g_1(\gamma, \pi_1 \underline{u}') : A(\gamma)$. Moreover $\pi_2 \underline{u}' : B'(\mathtt{f}(\gamma), \pi_1 \underline{u}')$ hence $q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}' : B'(\mathtt{f}(\gamma), f_1(\gamma, g_1(\gamma, \pi_1 \underline{u}')))$. Therefore:

$$g_2(\gamma, g_1(\gamma, \pi_1 \underline{u}'), q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}') : B(\gamma, g_1(\gamma, \pi_1 \underline{u}'))$$

hence:

$$\lfloor \gamma, \underline{u}' \rfloor \; g^{\Sigma}(\gamma, \underline{u}') \equiv \langle g_1(\gamma, \pi_1 \underline{u}'), g_2(\gamma, g_1(\gamma, \pi_1 \underline{u}'), q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}') \rangle : \Sigma_{x:A(\gamma)} B(\gamma, x).$$

We claim that $(f^{\Sigma}; g^{\Sigma})$ is a homotopy equivalence between:

$$\Sigma_{x:A(\gamma)} B(\gamma, x) \text{ and } \Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$$

relative to $(\mathtt{f}; \mathtt{g})$. In order to verify that $g^{\Sigma}(\gamma, \underline{u}') = u$, where $\underline{u}' \equiv f^{\Sigma}(\gamma, u)$, it is enough to verify that:

$$p : g_1(\gamma, \pi_1 \underline{u}') = \pi_1 u$$
$$g_2(\gamma, g_1(\gamma, \pi_1 \underline{u}'), q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}') = p^* \pi_2 u$$

as this implies that $g^{\Sigma}(\gamma, \underline{u}') = \langle \pi_1 u, \pi_2 u \rangle = u$. Since:

$$g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))) \bullet p_1(\gamma, \pi_1 u) : g_1(\gamma, \pi_1 \underline{u}') = \pi_1 u$$

we are left to verify that:

$$g_2(\gamma, g_1(\gamma, \pi_1 \underline{u}'), q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}') = p^* \pi_2 u$$

for $p \equiv g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))) \bullet p_1(\gamma, \pi_1 u)$. Let us observe that the following square:

$$\begin{array}{ccc}
f_1(\gamma, g_1(\gamma, \pi_1 \underline{u}')) & \xRightarrow{f_1(\gamma, g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))))} & f_1(\gamma, g_1(\gamma, f_1(\gamma, \pi_1 u))) \\
{\scriptstyle q_1(\gamma, \pi_1 \underline{u}')} \Downarrow & & \Downarrow {\scriptstyle q_1(\gamma, f_1(\gamma, \pi_1 u))} \\
\pi_1 \underline{u}' & \xRightarrow{\beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))} & f_1(\gamma, \pi_1 u)
\end{array}$$

commutes propositionally, because $\lfloor \gamma : \Gamma \rfloor\ q_1(\gamma, -) : f_1(\gamma, g_1(\gamma, -)) = -$ is a homotopy. Therefore:

$$\begin{aligned} q_1(\gamma, \pi_1\underline{u}') &= f_1(\gamma, g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u)))) \bullet q_1(\gamma, f_1(\gamma, \pi_1 u)) \\ &\quad \bullet \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))^{-1} \\ &= f_1(\gamma, g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u)))) \bullet f_1(\gamma, p_1(\gamma, \pi_1 u)) \\ &\quad \bullet \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))^{-1} \\ &= f_1(\gamma, g_1(\gamma, \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))) \bullet p_1(\gamma, \pi_1 u)) \\ &\quad \bullet \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))^{-1} \\ &\equiv f_1(\gamma, p) \bullet \beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))^{-1} \end{aligned}$$

where the second equality follows because $q_1(\gamma, f_1(\gamma, \pi_1 u)) = f_1(\gamma, p_1(\gamma, \pi_1 u))$ without loss of generality and the third by propositional functoriality. By propositional functoriality and groupoidality, we deduce that:

$$\begin{aligned} q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}' &= f_1(\gamma, p)^*(\beta_1(f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u))^{-1})^*\pi_2\underline{u}' \\ &= f_1(\gamma, p)^* f_2(\gamma, \pi_1 u, \pi_2 u) \end{aligned}$$

hence:

$$g_2(\gamma, g_1(\gamma, \pi_1\underline{u}'), q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}') = g_2(\gamma, g_1(\gamma, \pi_1\underline{u}'), f_1(\gamma, p)^* f_2(\gamma, \pi_1 u, \pi_2 u)).$$

Therefore we are left to observe that:

$$g_2(\gamma, g_1(\gamma, \pi_1\underline{u}'), f_1(\gamma, p)^* f_2(\gamma, \pi_1 u, \pi_2 u)) = p^*\pi_2 u$$

which follows by the judgement:

$$\lfloor p : x_1 = x_2, y : B(\gamma, x_2) \rfloor\ g_2(\gamma, x_1, f_1(\gamma, p)^* f_2(\gamma, x_2, y)) = p^* y$$

with $x_1 \equiv g_1(\gamma, \pi_1\underline{u}')$, $x_2 \equiv \pi_1 u$ and $y \equiv \pi_2 u$. This last judgement is true by path elimination on p and since $p_2(\gamma, x_1, y) : g_2(\gamma, x_1, f_2(\gamma, x_1, y)) = y$.

Vice versa, in order to verify that $f^\Sigma(\gamma, u) = \underline{u}'$, where $u \equiv g^\Sigma(\gamma, \underline{u}')$, let us observe that:

$$q \equiv f_1(\gamma, \beta_1(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}'))) \bullet q_1(\gamma, \pi_1\underline{u}') : f_1(\gamma, \pi_1 u) = \pi_1\underline{u}'$$

where:

$$\begin{aligned} \overline{g}_1(\gamma, \underline{u}') &\equiv g_1(\gamma, \pi_1\underline{u}') \\ \overline{g}_2(\gamma, \underline{u}') &\equiv g_2(\gamma, g_1(\gamma, \pi_1\underline{u}'), q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}'). \end{aligned}$$

Hence we are left to verify that:

$$f_2(\gamma, \pi_1 u, \pi_2 u) = q^*\pi_2\underline{u}'$$

as this implies that $f^\Sigma(\gamma, u) = \langle \pi_1\underline{u}', \pi_2\underline{u}' \rangle = \underline{u}'$. This is the case, as:

$$\begin{aligned} f_2(\gamma, \pi_1 u, \pi_2 u) &= f_1(\gamma, \beta_1(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')))^* f_2(\gamma, \overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')) \\ &= f_1(\gamma, \beta_1(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')))^* q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}' \\ &= (\ f_1(\gamma, \beta_1(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}'))) \bullet q_1(\gamma, \pi_1\underline{u}')\ )^*\pi_2\underline{u}' \\ &\equiv q^*\pi_2\underline{u}' \end{aligned}$$

where the second equality holds by propositional functoriality and since:

$$q_2(\gamma, g_1(\gamma, \pi_1\underline{u}'), q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}') : f_2(\gamma, \overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')) = q_1(\gamma, \pi_1\underline{u}')^*\pi_2\underline{u}'$$

and the third by propositional functoriality: we are left to verify the first. Since:

$$\lfloor x_1, x_2 : A(\gamma), y_1 : B(\gamma, x_1), y_2 : B(\gamma, x_2), p : x_1 = x_2, q : y_1 = p^* y_2 \rfloor$$
$$f_2(\gamma, x_1, y_1) = f_1(\gamma, p)^* f_2(\gamma, x_2, y_2)$$

by path elimination on p and q, we are done if:

$$p \equiv \beta_1(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')) \text{ and } q \equiv \beta_2(\overline{g}_1(\gamma, \underline{u}'), \overline{g}_2(\gamma, \underline{u}')).$$

Let us summarise the present subsection into the following:

Lemma 3.4. *Let us assume that $\gamma : \Gamma$ and $\delta : \Delta$ are contexts and that $(\mathtt{f};\mathtt{g})$ is a context homotopy equivalence between them. Moreover, let us assume that we are given judgements:*

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ *and* $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ *and* $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$

together with a homotopy equivalence $(f_1; g_1)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$, and a homotopy equivalence $(f_2; g_2)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2).

In context:

$$\gamma : \Gamma \text{ and } u : \Sigma_{x:A(\gamma)} B(\gamma, x) \text{ and } \underline{u}' : [\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma))$$

if we name:

$$\lfloor \gamma, u \rfloor\ f^{\Sigma}(\gamma, u) \equiv \langle f_1(\gamma, \pi_1 u), f_2(\gamma, \pi_1 u, \pi_2 u) \rangle : [\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}(\gamma))$$
$$\lfloor \gamma, \underline{u}' \rfloor\ g^{\Sigma}(\gamma, \underline{u}') \equiv \langle g_1(\gamma, \pi_1 \underline{u}'), g_2(\gamma, g_1(\gamma, \pi_1 \underline{u}'), q_1(\gamma, \pi_1 \underline{u}')^* \pi_2 \underline{u}') \rangle : \Sigma_{x:A(\gamma)} B(\gamma, x)$$

then $(f^{\Sigma}; g^{\Sigma})$ is a homotopy equivalence between $\Sigma_{x:A(\gamma)} B(\gamma, x)$ and $\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ relative to $(\mathtt{f};\mathtt{g})$.

3.4. **Identity types over homotopy equivalent types.** Again, let us describe some:

Data. Let us assume that we are given a context homotopy equivalence $(\mathtt{f};\mathtt{g})$ between $\gamma : \Gamma$ and $\delta : \Delta$ and judgements:

$$\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \text{ and } \lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$$

together with a homotopy equivalence $(f; g)$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$. Moreover let us assume that we are given judgements:

$$\lfloor \gamma : \Gamma \rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma)$$
$$\lfloor \delta : \Delta \rfloor\ t_1(\delta), t_2(\delta) : A'(\delta)$$

together with:

$$\lfloor \gamma : \Gamma \rfloor\ \mathsf{r}_1(\gamma) : f(\gamma, s_1(\gamma)) = t_1(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma \rfloor\ \mathsf{r}_2(\gamma) : f(\gamma, s_2(\gamma)) = t_2(\mathtt{f}(\gamma)).$$

Fact. We can define a homotopy equivalence $(f^{=}; g^{=})$ between $s_1(\gamma) = s_2(\gamma)$ and $t_1(\delta) = t_2(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ as follows, leading to Lemma 3.5 below.

Let us consider homotopies:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor \; p(\gamma, x) : g(\gamma, f(\gamma, x)) = x$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor \; q(\gamma, \underline{x}') : f(\gamma, g(\gamma, \underline{x}')) = \underline{x}'$$

We define:

$$\lfloor \gamma : \Gamma, p : s_1(\gamma) = s_2(\gamma) \rfloor \; f^{=}(\gamma, p) \equiv \mathsf{r}_1(\gamma)^{-1} \bullet f(\gamma, p) \bullet \mathsf{r}_2(\gamma) : t_1(\mathtt{f}(\gamma)) = t_2(\mathtt{f}(\gamma))$$

and:

$$\lfloor \gamma : \Gamma, \underline{p}' : t_1(\mathtt{f}(\gamma)) = t_2(\mathtt{f}(\gamma)) \rfloor$$
$$g^{=}(\gamma, \overline{p}') \equiv p(\gamma, s_1(\gamma))^{-1} \bullet g(\gamma, \mathsf{r}_1(\gamma) \bullet \underline{p}' \bullet \mathsf{r}_2(\gamma)^{-1}) \bullet p(\gamma, s_2(\gamma)) : s_1(\gamma) = s_2(\gamma).$$

Let us observe that:

$$g^{=}(\gamma, f^{=}(\gamma, p)) = p(\gamma, s_1(\gamma))^{-1} \bullet g(\gamma, f(\gamma, p)) \bullet p(\gamma, s_2(\gamma)) = p$$

where the first identity type is inhabited by groupoidality and the second since:

$$\lfloor \gamma : \Gamma \rfloor \; p(\gamma, -) : g(\gamma, f(\gamma, -)) = -.$$

Moreover, by propositional functoriality:

$$f^{=}(\gamma, g^{=}(\gamma, \underline{p}')) = a(\gamma) \bullet f(\gamma, g(\gamma, \underline{p}')) \bullet b(\gamma)$$

where:

$$\begin{aligned} a(\gamma) &\equiv \mathsf{r}_1(\gamma)^{-1} \bullet f(\gamma, p(\gamma, s_1(\gamma)))^{-1} \bullet f(\gamma, g(\gamma, \mathsf{r}_1(\gamma))) \\ &= \mathsf{r}_1(\gamma)^{-1} \bullet q(\gamma, f(\gamma, s_1(\gamma)))^{-1} \bullet f(\gamma, g(\gamma, \mathsf{r}_1(\gamma))) \\ b(\gamma) &\equiv f(\gamma, g(\gamma, \mathsf{r}_2(\gamma)))^{-1} \bullet f(\gamma, p(\gamma, s_2(\gamma))) \bullet \mathsf{r}_2(\gamma) \\ &= f(\gamma, g(\gamma, \mathsf{r}_2(\gamma)))^{-1} \bullet q(\gamma, f(\gamma, s_2(\gamma))) \bullet \mathsf{r}_2(\gamma) \end{aligned}$$

where the identity types are inhabited because $\lfloor \gamma, x \rfloor \; f(\gamma, p(\gamma, x)) = q(\gamma, f(\gamma, x))$ without loss of generality. Observe that:

$$a(\gamma) = q(\gamma, t_1(\mathtt{f}(\gamma)))^{-1}$$
$$b(\gamma) = q(\gamma, t_2(\mathtt{f}(\gamma)))$$

as the diagram:

$$\begin{array}{ccc} f(\gamma, s_i(\gamma)) & \xRightarrow{\;\mathsf{r}_i(\gamma)\;} & t_i(\mathtt{f}(\gamma)) \\ {\scriptstyle q(\gamma, f(\gamma, s_i(\gamma)))}\Uparrow & & \Uparrow{\scriptstyle q(\gamma, t_i(\mathtt{f}(\gamma)))} \\ f(\gamma, g(\gamma, f(\gamma, s_i(\gamma)))) & \xRightarrow{\;f(\gamma, g(\gamma, \mathsf{r}_i(\gamma)))\;} & f(\gamma, g(\gamma, t_i(\mathtt{f}(\gamma)))) \end{array}$$

commutes propositionally for $i = 1, 2$, since $\lfloor \gamma : \Gamma \rfloor \; q(\gamma, -) : f(\gamma, g(\gamma, -)) = -$. We conclude that:

$$f^{=}(\gamma, g^{=}(\gamma, \underline{p}')) = q(\gamma, t_1(\mathtt{f}(\gamma)))^{-1} \bullet f(\gamma, g(\gamma, \underline{p}')) \bullet q(\gamma, t_2(\mathtt{f}(\gamma))) = \underline{p}'$$

by groupoidality and since $\lfloor \gamma : \Gamma \rfloor \; q(\gamma, -) : f(\gamma, g(\gamma, -)) = -$. We summarise this fact into the following:

Lemma 3.5. *Let us assume that we are given a context homotopy equivalence* $(\mathtt{f};\mathtt{g})$ *between* $\gamma:\Gamma$ *and* $\delta:\Delta$ *and judgements:*

$$\lfloor \gamma:\Gamma \rfloor\ A(\gamma):\textsc{Type} \text{ and } \lfloor \delta:\Delta \rfloor\ A'(\delta):\textsc{Type}$$

together with a homotopy equivalence $(f;g)$*:*

$$\lfloor \gamma:\Gamma, x:A(\gamma) \rfloor\ f(\gamma,x):A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma:\Gamma, \underline{x}':A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma,\underline{x}'):A(\gamma)$$

between $A(\gamma)$ *and* $A'(\delta)$ *relative to* $(\mathtt{f};\mathtt{g})$*. Moreover let us assume that we are given judgements:*

$$\lfloor \gamma:\Gamma \rfloor\ s_1(\gamma), s_2(\gamma):A(\gamma)$$
$$\lfloor \delta:\Delta \rfloor\ t_1(\delta), t_2(\delta):A'(\delta)$$

together with:

$$\lfloor \gamma:\Gamma \rfloor\ \mathsf{r}_1(\gamma): f(\gamma, s_1(\gamma)) = t_1(\mathtt{f}(\gamma))$$
$$\lfloor \gamma:\Gamma \rfloor\ \mathsf{r}_2(\gamma): f(\gamma, s_2(\gamma)) = t_2(\mathtt{f}(\gamma)).$$

If we name:

$$\lfloor \gamma:\Gamma, p:s_1(\gamma)=s_2(\gamma) \rfloor\ f^{=}(\gamma,p) \equiv \mathsf{r}_1(\gamma)^{-1} \bullet f(\gamma,p) \bullet \mathsf{r}_2(\gamma): t_1(\mathtt{f}(\gamma)) = t_2(\mathtt{f}(\gamma))$$
$$\lfloor \gamma:\Gamma, \underline{p}':t_1(\mathtt{f}(\gamma)) = t_2(\mathtt{f}(\gamma)) \rfloor\ g^{=}(\gamma,\underline{p}') \equiv$$
$$p(\gamma,s_1(\gamma))^{-1} \bullet g(\gamma, \mathsf{r}_1(\gamma) \bullet \underline{p}' \bullet \mathsf{r}_2(\gamma)^{-1}) \bullet p(\gamma, s_2(\gamma)) : s_1(\gamma)=s_2(\gamma)$$

then $(f^{=};g^{=})$ *is a homotopy equivalence between* $s_1(\gamma)=s_2(\gamma)$ *and* $t_1(\delta)=t_2(\delta)$ *relative to* $(\mathtt{f};\mathtt{g})$*.*

3.5. **Canonical homotopy equivalences between types.** In Subsection 3.6 we are going to use Lemma 3.2 to define a specific class of context homotopy equivalences. However, in order to do so, we first need to identify a particular notion of homotopy equivalence between types relative to a given context homotopy equivalence: we are going to apply Lemma 3.2 to these ones only. In this subsection we inductively present this notion.

Let $\gamma:\Gamma$ and $\delta:\Delta$ be contexts together with a context homotopy equivalence $(\mathtt{f};\mathtt{g})$ and let $\lfloor \gamma:\Gamma \rfloor\ S(\gamma):\textsc{Type}$ and $\lfloor \delta:\Delta \rfloor\ T(\delta):\textsc{Type}$ have h-propositional identities. We provide a list of inductive clauses determining the family of the **canonical homotopy equivalences** $\langle\, \varphi \mid \psi \,\rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f};\mathtt{g})$:

(a) If $S(\gamma)\equiv S$ and $T(\delta)\equiv S(\gamma)\equiv S$, then the identity of S (as judgement of the form $\lfloor \gamma:\Gamma, s:S \rfloor\ s:S$) and itself constitute a canonical homotopy equivalence $\langle\, \varphi \mid \psi \,\rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f};\mathtt{g})$. Observe that this homotopy equivalence between S and itself is in fact *relative to* $(\mathtt{f};\mathtt{g})$, as:
 - $\lfloor \gamma:\Gamma, s:S \rfloor\ \varphi(\gamma,s)\equiv s:S\equiv S(\mathtt{f}(\gamma))$
 - $\lfloor \gamma:\Gamma, s:S\equiv S(\mathtt{f}(\gamma)) \rfloor\ \psi(\gamma,s)\equiv s:S$.

(b) If $S(\gamma)\equiv \Pi_{x:A(\gamma)}B(\gamma,x)$ and $T(\delta)\equiv \Pi_{\underline{x}:A'(\delta)}B'(\delta,\underline{x})$ for some judgements:
 - $\lfloor \gamma:\Gamma \rfloor\ A(\gamma):\textsc{Type}$ and $\lfloor \gamma:\Gamma, x:A(\gamma) \rfloor\ B(\gamma,x):\textsc{Type}$
 - $\lfloor \delta:\Delta \rfloor\ A'(\delta):\textsc{Type}$ and $\lfloor \delta:\Delta, \underline{x}:A'(\delta) \rfloor\ B'(\delta,\underline{x}):\textsc{Type}$

 and if there are a canonical homotopy equivalence $\langle\, f_1 \mid g_1 \,\rangle$:

$$\lfloor \gamma:\Gamma, x:A(\gamma) \rfloor\ f_1(\gamma,x):A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma:\Gamma, \underline{x}':A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma,\underline{x}'):A(\gamma)$$

 between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$ and a canonical homotopy equivalence $\langle\, f_2 \mid g_2 \,\rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f}; \mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2), then the homotopy equivalence $(f^{\Pi}; g^{\Pi})$ of Lemma 3.3 is a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$.

(c) If $S(\gamma) \equiv \Sigma_{x:A(\gamma)} B(\gamma, x)$ and $T(\delta) \equiv \Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ for some judgements:
– $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}$
– $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$
and if there are a canonical homotopy equivalence $\langle\ f_1 \mid g_1\ \rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and a canonical homotopy equivalence $\langle\ f_2 \mid g_2\ \rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f}; \mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2), then the homotopy equivalence $(f^{\Sigma}; g^{\Sigma})$ of Lemma 3.4 is a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$.

(d) If $S(\gamma) \equiv s_1(\gamma) = s_2(\gamma)$ and $T(\delta) \equiv t_1(\delta) = t_2(\delta)$ for some judgements:

$$\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \quad \text{and} \quad \lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$$

and some judgements:

$$\lfloor \gamma : \Gamma \rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma) \quad \text{and} \quad \lfloor \delta : \Delta \rfloor\ t_1(\delta), t_2(\delta) : A'(\delta)$$

and if there are a canonical homotopy equivalence $\langle\ f \mid g\ \rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and judgements:

$$\lfloor \gamma : \Gamma \rfloor\ \mathsf{r}_1(\gamma) : f(\gamma, s_1(\gamma)) = t_1(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma \rfloor\ \mathsf{r}_2(\gamma) : f(\gamma, s_2(\gamma)) = t_2(\mathtt{f}(\gamma)).$$

then the homotopy equivalence $(f^{=}, g^{=})$ of Lemma 3.5 is a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$.

3.6. **Canonical homotopy equivalences between contexts.** In this subsection we recursively define a family of context homotopy equivalences that we refer to as the *canonical* ones. In fact, when we make the syntax of PTT into a model of ETT, we need to define equivalence classes of contexts modulo the context homotopy equivalences of this particular shape that we are going to identify.

As mentioned before, this proof strategy, based on defining a family of canonical equivalences between contexts—up to which contexts are identified—appears in a work by Hofmann [Hof95a, Hof96] in order to obtain an analogous conservativity result. Moreover, such a notion of canonical equivalence appears—in a similar formulation—in a work by Maietti [Mai09] in order to define an interpretation of the extensional level of Minimalist Foundation within the intensional one, as well as in one by Contente and Maietti [CM24] in order to interpret the former within Homotopy Type Theory, and in one by Maietti and Sabelli [MS25] to define an interpretation within the former of itself extendend with an extensionality axiom for its propositions.

Let $\gamma : \Gamma$ and $\delta : \Delta$ be two contexts. We give a list of inductive clauses determining the family of the **canonical context homotopy equivalences** $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between them:

(1) If $\gamma : \Gamma$ and $\delta : \Delta$ are the empty context then the empty list (as context morphism $\Gamma \to \Delta$) and itself (as context morphism $\Delta \to \Gamma$) constitute a canonical context homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$.

(2) If $\gamma : \Gamma$ and $\delta : \Delta$ are of the form $\gamma' : \Gamma', x : A(\gamma')$ and $\delta' : \Delta', \underline{x} : A'(\delta')$ respectively and:

$$\lfloor \gamma' : \Gamma' \rfloor\ \mathtt{f}(\gamma') : \Delta'$$
$$\lfloor \delta' : \Delta' \rfloor\ \mathtt{g}(\delta') : \Gamma'$$

is a canonical context homotopy equivalence $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ and:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor\ f(\gamma', x) : A'(\mathtt{f}(\gamma'))$$
$$\lfloor \gamma' : \Gamma', \underline{x}' : A'(\mathtt{f}(\gamma')) \rfloor\ g(\gamma', \underline{x}') : A(\gamma')$$

is a canonical homotopy equivalence $\langle\ f \mid g\ \rangle$ between $A(\gamma')$ and $A'(\delta')$ relative to $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ (in the sense of Subsection 3.5) then the extension:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor\ (\mathtt{f}, \mathtt{f}_{m+1})(\gamma', x) : \Delta', A'$$
$$\lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})(\delta', \underline{x}) : \Gamma', A$$

of $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ via $\langle\ f \mid g\ \rangle$ (Lemma 3.2) is a canonical context homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$.

We end the current section with the following:

Remark 3.6. Let $A : \textsc{Type}$, $a, b : A$, and $q : a = b$ be atomic judgements of PTT. By (a) of Subsection 3.5, the pair:

$$\langle\ \ \lfloor x : A \rfloor\ x : A \ \ \mid\ \ \lfloor x : A \rfloor\ x : A\ \ \rangle$$

constitutes a canonical homotopy equivalence between A and itself relative to the canonical context homotopy equivalence of (1). Therefore, by (b) of Subsection 3.5, the pair:

$$\langle\ \ \lfloor p : a = a \rfloor\ \mathsf{r}(a)^{-1} \bullet p \bullet q : a = b\ \ \mid\ \ \lfloor p' : a = b \rfloor\ \mathsf{r}(a) \bullet p' \bullet q^{-1} : a = a\ \ \rangle$$

constitutes a canonical homotopy equivalence between the types $a = a$ and $a = b$ relative to the canonical context homotopy equivalence of (1). By (2) we obtain a canonical context homotopy equivalence:

$$\langle\ \ \lfloor p : a = a \rfloor\ \mathsf{r}(a)^{-1} \bullet p \bullet q : a = b\ \ \mid\ \ \lfloor p' : a = b \rfloor\ \mathsf{r}(a) \bullet p' \bullet q^{-1} : a = a\ \ \rangle$$

between the context $p : a = a$ and the context $p' : a = b$.

However, assuming that $a \equiv b$ (and w.l.o.g. that $p' \equiv p$) this homotopy equivalence is homotopic—i.e. propositionally equal—to the identity over the context $p : a = a$ if and only if the type $q = \mathsf{r}(a)$ is inhabited—cf. Proposition 4.17 on contexts with h-propositional identities.

We will use this fundamental canonical context homotopy equivalence, which we have presented here in a particular case, in Subsection 6.2 to show that the syntax of PTT with contexts identified modulo these canonical equivalences constitutes a model of extensional identity types.

4. Properties of the canonical homotopy equivalences

In this section we prove those properties of the family of the canonical context homotopy equivalences (defined in Subsection 3.6) that are needed in order to define a model of ETT starting from the syntax of PTT. Our approach is the following: we start by proving properties (e.g. the fact that an equivalence relation is induced between the contexts of PTT) for the general family of canonical equivalences; secondly, we restrict to a smaller family of contexts, called *contexts with h-propositional identities*, and we only consider canonical equivalences between them, so that additional properties (e.g. the uniqueness—up to homotopy—of a canonical equivalence between two given contexts) are satisfied.

4.1. **Properties of the family of canonical equivalences.** We remind that we use the notation $(\,\cdot\,;\,\cdot\,)$ to indicate any homotopy equivalence between contexts or types in context, reserving the symbol $\langle\,\cdot\mid\cdot\,\rangle$ for the canonical ones. We start our list of results from the following observation:

Lemma 4.1. *Let $\gamma:\Gamma$ be a context $\gamma_1:\Gamma_1,\gamma_2:\Gamma_2(\gamma_1),...,\gamma_n:\Gamma_n(\gamma_1,...,\gamma_{n-1})$ and let $\delta:\Delta$ be a context $\delta_1:\Delta_1,\delta_2:\Delta_2(\delta_1),...,\delta_m:\Delta_m(\delta_1,...,\delta_{m-1})$. If there is a canonical context homotopy equivalence $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$ between $\gamma:\Gamma$ and $\delta:\Delta$ then $n=m$.*

Proof. By induction on the complexity of $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$. If $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$ is of the form (1) then $n=0=m$ and we are done. If $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$ is of the form (2) for some canonical context homotopy equivalence $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$ and some canonical homotopy equivalence $\langle\,f\mid g\,\rangle$ relative to $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$, then by inductive hypothesis $n-1=m-1$ hence we are done. □

Secondly, we prove the properties of reflexivity, symmetry and transitivity of the relation between contexts induced by the family of the canonical equivalences between them.

Proposition 4.2 (Reflexivity)**.** *Let $\gamma:\Gamma$ be a context:*

$$\gamma_1:\Gamma_1,\gamma_2:\Gamma_2(\gamma_1),...,\gamma_n:\Gamma_n(\gamma_1,...,\gamma_{n-1}).$$

Then there is a canonical context homotopy equivalence $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$ from γ to γ such that $\lfloor\gamma\rfloor\ \mathtt{c}(\gamma)=\gamma$ (i.e. $\lfloor\gamma\rfloor\ \mathtt{d}(\gamma)=\gamma$).

Proof. By induction on the length n of $\gamma:\Gamma$. If $n=0$ then we are done by (1). Otherwise, let $\gamma':\Gamma'$ be the context $\gamma_1,\gamma_2,...,\gamma_{n-1}$. By inductive hypothesis, there is a canonical context homotopy equivalence $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$ between γ' and itself such that $\lfloor\gamma'\rfloor\ \alpha(\gamma'):\mathtt{f}(\gamma')=\gamma'$ (i.e. $\lfloor\gamma'\rfloor\ \mathtt{g}(\gamma')=\gamma'$).

Let us pretend that we know that there is a canonical homotopy equivalence $\langle\,f\mid g\,\rangle$ between $\Gamma_n(\gamma')$ and itself relative to $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$ and such that $\lfloor\gamma',\gamma_n:\Gamma_n(\gamma')\rfloor\ f(\gamma',\gamma_n)=\alpha(\gamma')^*\gamma_n$ (i.e. $\lfloor\gamma',\gamma_n\rfloor\ g(\gamma',\alpha(\gamma')^*\gamma_n)=\gamma_n$). Then the extension:

$$\lfloor\gamma':\Gamma',\gamma_n:\Gamma_n(\gamma')\rfloor\ (\mathtt{f},\mathtt{f}_{m+1})(\gamma',\gamma_n):\Gamma',\Gamma_n$$
$$\lfloor\gamma':\Gamma',\gamma_n:\Gamma_n(\gamma')\rfloor\ (\mathtt{g},\mathtt{g}_{n+1})(\gamma',\gamma_n):\Gamma',\Gamma_n$$

of $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$ via $\langle\,f\mid g\,\rangle$ (Lemma 3.2) satisfies $(\mathtt{f},\mathtt{f}_{m+1})(\gamma)=\gamma$ (hence $(\mathtt{g},\mathtt{g}_{n+1})(\gamma)=\gamma$). Moreover, it is a canonical homotopy equivalence $\langle\,\mathtt{c}\mid\mathtt{d}\,\rangle$ by (2).

Hence we are done if there is a canonical homotopy equivalence $\langle\,f\mid g\,\rangle$ between $\Gamma_n(\gamma')$ and itself relative to $\langle\,\mathtt{f}\mid\mathtt{g}\,\rangle$ and such that $\lfloor\gamma',\gamma_n:\Gamma_n(\gamma')\rfloor\ f(\gamma',\gamma_n)=\alpha(\gamma')^*\gamma_n$. But this is true because of the following Lemma 4.3. □

Lemma 4.3. *Let $\gamma : \Gamma$ be a context and let* $(\mathtt{f};\mathtt{g})$ *be a context homotopy equivalence between γ and itself such that $\lfloor\gamma\rfloor\ \alpha(\gamma) : \mathtt{f}(\gamma) = \gamma$ (and equivalently $\lfloor\gamma\rfloor\ \mathtt{g}(\gamma) = \gamma$). Moreover, let us assume that $\lfloor\gamma : \Gamma\rfloor\ S(\gamma) : \textsc{Type}$. Then there exists a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between $S(\gamma)$ and itself relative to* $(\mathtt{f};\mathtt{g})$ *and such that:*

$$\lfloor\gamma, s : S(\gamma)\rfloor\ \varphi(\gamma, s) = \alpha(\gamma)^* s\ \text{ (i.e. } \lfloor\gamma, s\rfloor\ \psi(\gamma, \alpha(\gamma)^* s) = s\text{).}$$

Proof. By induction on the complexity of the type $S(\gamma)$.

If $S(\gamma) \equiv S$ then by (a) the pair:

$$(\lfloor\gamma, s : S\rfloor\ \varphi(\gamma, s) \equiv s : S;\ \lfloor\gamma, s : S\rfloor\ \psi(\gamma, s) \equiv s : S)$$

is a canonical homotopy equivalence between S and itself relative to $(\mathtt{f};\mathtt{g})$. Moreover, let us observe that:

$$\lfloor\gamma, s\rfloor\ \varphi(\gamma, s) \equiv s = \alpha(\gamma)^* s$$

—by (generalised) path induction on $\alpha(\gamma)$—hence we are done.

If $S(\gamma)$ is of the form $\Pi_{x:A(\gamma)}B(\gamma, x)$ for some judgements $\lfloor\gamma : \Gamma\rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor\gamma : \Gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \textsc{Type}$, then by inductive hypothesis there are a canonical homotopy equivalence $\langle\ f_1 \mid g_1\ \rangle$:

$$\lfloor\gamma : \Gamma, x : A(\gamma)\rfloor\ f_1(\gamma, x) : A(\mathtt{f}(\gamma))$$
$$\lfloor\gamma : \Gamma, x' : A(\mathtt{f}(\gamma))\rfloor\ g_1(\gamma, x') : A(\gamma)$$

between $A(\gamma)$ and itself relative to $(\mathtt{f};\mathtt{g})$ such that $\lfloor\gamma, x\rfloor\ \alpha_1(\gamma, x) : f_1(\gamma, x) = \alpha(\gamma)^* x$ and equivalently $\lfloor\gamma, x\rfloor\ g_1(\gamma, \alpha(\gamma)^* x) = x$ and a canonical homotopy equivalence $\langle\ f_2 \mid g_2\ \rangle$:

$$\lfloor\gamma : \Gamma, x : A(\gamma), y : B(\gamma, x)\rfloor\ f_2(\gamma, x, y) : B(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor\gamma : \Gamma, x : A(\gamma), y' : B(\mathtt{f}(\gamma), f_1(\gamma, x))\rfloor\ g_2(\gamma, x, \underline{y'}) : B(\gamma, x)$$

between $B(\gamma, x)$ and itself relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2) such that $\lfloor\gamma, x, y\rfloor\ f_2(\gamma, x, y) = (\alpha(\gamma), \alpha_1(\gamma, x))^* y$ and equivalently:

$$\lfloor\gamma, x, y\rfloor\ g_2(\gamma, x, (\alpha(\gamma), \alpha_1(\gamma, x))^* y) = y.$$

The homotopy equivalence $(f^\Pi; g^\Pi)$ of Lemma 3.3 is a canonical homotopy equivalence relative to $(\mathtt{f};\mathtt{g})$. We are left to verify that:

$$\lfloor\gamma, z : \Pi_{x:A(\gamma)}B(\gamma, x)\rfloor\ g^\Pi(\gamma, \alpha(\gamma)^* z) = z.$$

Let us observe that:

$$\lfloor\gamma, z, x\rfloor\ \mathsf{ev}(\alpha(\gamma)^* z, f_1(\gamma, x)) = \mathsf{ev}(\alpha(\gamma)^* z, \alpha(\gamma)^* x) = (\alpha(\gamma), \alpha_1(\gamma, x))^* \mathsf{ev}(z, x)$$

where the second equality holds by multiple (generalised) path induction on $(\alpha(\gamma), \alpha_1(\gamma, x))$. Hence, by propositional functoriality:

$$\lfloor\gamma, z, x\rfloor\ g_2(\gamma, x, \mathsf{ev}(\alpha(\gamma)^* z, f_1(\gamma, x))) = g_2(\gamma, x, (\alpha(\gamma), \alpha_1(\gamma, x))^* \mathsf{ev}(z, x)) = \mathsf{ev}(z, x)$$

where the second equality holds by substitution into the judgement:

$$\lfloor\gamma, x, y\rfloor\ g_2(\gamma, x, (\alpha(\gamma), \alpha_1(\gamma, x))^* y) = y.$$

Finally, by propositional functoriality and propositional η-expansion:

$$\lfloor\gamma, z\rfloor\ g^\Pi(\gamma, \alpha(\gamma)^* z) \equiv \lambda x.g_2(\gamma, x, \mathsf{ev}(\alpha(\gamma)^* z, f_1(\gamma, x))) = \lambda x.\mathsf{ev}(z, x) = z$$

and we are done.

If $S(\gamma)$ is of the form $\Sigma_{x:A(\gamma)}B(\gamma,x)$ for some judgements $\lfloor\gamma:\Gamma\rfloor\ A(\gamma):\textsc{Type}$ and $\lfloor\gamma:\Gamma,x:A(\gamma)\rfloor\ B(\gamma,x):\textsc{Type}$, then by inductive hypothesis there are a canonical homotopy equivalence $\langle\ f_1 \mid g_1\ \rangle$:

$$\lfloor\gamma:\Gamma,x:A(\gamma)\rfloor\ f_1(\gamma,x):A(\mathtt{f}(\gamma))$$
$$\lfloor\gamma:\Gamma,x':A(\mathtt{f}(\gamma))\rfloor\ g_1(\gamma,x'):A(\gamma)$$

between $A(\gamma)$ and itself relative to $(\mathtt{f};\mathtt{g})$ such that $\lfloor\gamma,x\rfloor\ \alpha_1(\gamma,x):f_1(\gamma,x)=\alpha(\gamma)^*x$ and equivalently $\lfloor\gamma,x\rfloor\ g_1(\gamma,\alpha(\gamma)^*x)=x$ and a canonical homotopy equivalence $\langle\ f_2 \mid g_2\ \rangle$:

$$\lfloor\gamma:\Gamma,x:A(\gamma),y:B(\gamma,x)\rfloor\ f_2(\gamma,x,y):B(\mathtt{f}(\gamma),f_1(\gamma,x))$$
$$\lfloor\gamma:\Gamma,x:A(\gamma),y':B(\mathtt{f}(\gamma),f_1(\gamma,x))\rfloor\ g_2(\gamma,x,\underline{y'}):B(\gamma,x)$$

between $B(\gamma,x)$ and itself relative to the extension $(\mathtt{f},\mathtt{f}_{m+1};\mathtt{g},\mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1;g_1)$ (Lemma 3.2) such that $\lfloor\gamma,x,y\rfloor\ f_2(\gamma,x,y)=(\alpha(\gamma),\alpha_1(\gamma,x))^*y$ and equivalently:

$$\lfloor\gamma,x,y\rfloor\ g_2(\gamma,x,(\alpha(\gamma),\alpha_1(\gamma,x))^*y)=y.$$

The homotopy equivalence $(f^\Sigma;g^\Sigma)$ of Lemma 3.4 is a canonical homotopy equivalence relative to $(\mathtt{f};\mathtt{g})$, hence we are left to verify that:

$$\lfloor\gamma,u:\Sigma_{x:A(\gamma)}B(\gamma,x)\rfloor\ f^\Sigma(\gamma,u)\equiv\langle f_1(\gamma,\pi_1u),f_2(\gamma,\pi_1u,\pi_2u)\rangle=\alpha(\gamma)^*u.$$

As $\alpha_1(\gamma,\pi_1u):f_1(\gamma,\pi_1u)=\alpha(\gamma)^*\pi_1u$ and:

$$f_2(\gamma,\pi_1u,\pi_2u)=(\alpha(\gamma),\alpha_1(\gamma,\pi_1u))^*\pi_2u=\alpha_1(\gamma,\pi_1u)^*(\alpha(\gamma)^*\pi_2u)$$

—where the last equality follows by (generalised) path induction on $\alpha_1(\gamma,\pi_1u)$—it is in fact the case that:

$$\lfloor\gamma,u\rfloor\ f^\Sigma(\gamma,u)=\langle\alpha(\gamma)^*\pi_1u,\alpha(\gamma)^*\pi_2u\rangle=\alpha(\gamma)^*\langle\pi_1u,\pi_2u\rangle=\alpha(\gamma)^*u$$

where the second equality follows by multiple (generalised) path induction on $\alpha(\gamma)$ and the third by propositional functoriality and propositional η-expansion.

If $S(\gamma)$ is of the form $s_1(\gamma)=s_2(\gamma)$ for some judgement $\lfloor\gamma:\Gamma\rfloor\ A(\gamma):\textsc{Type}$, then by inductive hypothesis there is a canonical homotopy equivalence $\langle\ f \mid g\ \rangle$:

$$\lfloor\gamma:\Gamma,x:A(\gamma)\rfloor\ f(\gamma,x):A(\mathtt{f}(\gamma))$$
$$\lfloor\gamma:\Gamma,x':A(\mathtt{f}(\gamma))\rfloor\ g(\gamma,x'):A(\gamma)$$

between $A(\gamma)$ and itself relative to $(\mathtt{f};\mathtt{g})$ such that $\lfloor\gamma,x\rfloor\ \alpha_1(\gamma,x):f(\gamma,x)=\alpha(\gamma)^*x$ and equivalently $\lfloor\gamma,x\rfloor\ g_1(\gamma,\alpha(\gamma)^*x)=x$. For $i=1,2$ let us observe that:

$$\lfloor\gamma\rfloor\ \alpha_1(\gamma,s_i(\gamma)):f(\gamma,s_i(\gamma))=\alpha(\gamma)^*s_i(\gamma)\text{ and }\lfloor\gamma\rfloor\ s_i(\alpha(\gamma)):s_i(\mathtt{f}(\gamma))=\alpha(\gamma)^*s_i(\gamma)$$

hence:

$$\lfloor\gamma:\Gamma\rfloor\ r_i(\gamma)\equiv\alpha_1(\gamma,s_i(\gamma))\bullet s_i(\alpha(\gamma))^{-1}:f(\gamma,s_i(\gamma))=s_i(\mathtt{f}(\gamma))$$

and then the homotopy equivalence $(f^=,g^=)$ of Lemma 3.5 is a canonical homotopy equivalence relative to $(\mathtt{f};\mathtt{g})$. We are left to show that $\lfloor\gamma,p:s_1(\gamma)=s_2(\gamma)\rfloor\ f^=(\gamma,p)=\alpha(\gamma)^*p$. In fact:

$$\begin{aligned}\lfloor\gamma,p\rfloor\ f^=(\gamma,p)&=s_1(\alpha(\gamma))\bullet\alpha_1(\gamma,s_1(\gamma))^{-1}\bullet f(\gamma,p)\bullet\alpha_1(\gamma,s_2(\gamma))\bullet s_2(\alpha(\gamma))^{-1}\\&=s_1(\alpha(\gamma))\bullet\alpha(\gamma)^*p\bullet s_2(\alpha(\gamma))^{-1}\\&=\alpha(\gamma)^*p\end{aligned}$$

by propositional groupoidality and since $\lfloor \gamma, x \rfloor\ \alpha_1(\gamma, x) : f(\gamma, x) = \alpha(\gamma)^* x$. Here the last of these equalities follows by multiple (generalised) path induction on $\alpha(\gamma)$ and by propositional groupoidality. □

Proposition 4.4 (Symmetry)**.** *If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is a canonical context homotopy equivalence between $\gamma : \Gamma$ and $\delta : \Delta$, then there is a canonical context homotopy equivalence $\langle\ \mathtt{d}' \mid \mathtt{c}'\ \rangle$ between δ and γ such that $\lfloor \gamma : \Gamma \rfloor\ \mathtt{c}(\gamma) = \mathtt{c}'(\gamma)$ and equivalently $\lfloor \delta : \Delta \rfloor\ \mathtt{d}(\delta) = \mathtt{d}'(\delta)$.*

Proof. By induction on the complexity of $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form (1) then we are done. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form (2) for some canonical context homotopy equivalence $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ between $\gamma' : \Gamma'$ and $\delta' : \Delta'$ and some canonical homotopy equivalence $\langle\ f \mid g\ \rangle$ between $\lfloor \gamma' \rfloor\ A(\gamma')$ and $\lfloor \delta' \rfloor\ A'(\delta')$ relative to $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$

$$\text{—hence } \gamma \equiv \gamma', x : A(\gamma') \text{ and } \delta \equiv \delta', \underline{x} : A'(\delta')\text{—}$$

then by inductive hypothesis there is a canonical context homotopy equivalence $\langle\ \mathtt{g}' \mid \mathtt{f}'\ \rangle$ such that $\lfloor \gamma' : \Gamma' \rfloor\ \mathtt{f}(\gamma') = \mathtt{f}'(\gamma')$ and equivalently $\lfloor \delta' : \Delta' \rfloor\ \alpha(\delta') : \mathtt{g}(\delta') = \mathtt{g}'(\delta')$. Moreover, by the following Lemma 4.5 there is a canonical homotopy equivalence $\langle\ g' \mid f'\ \rangle$:

$$\begin{gathered} \lfloor \delta' : \Delta', \underline{x} : A'(\delta) \rfloor\ g'(\delta', \underline{x}) : A(\mathtt{g}'(\delta')) \\ \lfloor \delta' : \Delta', x' : A(\mathtt{g}'(\delta')) \rfloor\ f'(\delta', x') : A'(\delta') \end{gathered}$$

relative to $(\mathtt{g}'; \mathtt{f}')$ such that:

$$\begin{gathered} \lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor\ \alpha_1(\delta', \underline{x}) : g(\mathtt{g}(\delta'), \mathtt{q}(\delta')^* \underline{x}) = \alpha(\delta')^* g'(\delta', \underline{x}) \\ (\text{ i.e. } \lfloor \delta' : \Delta', x' : A(\mathtt{g}(\delta')) \rfloor\ f(\mathtt{g}(\delta'), \alpha(\delta')^* x') = \mathtt{q}(\delta')^* f'(\delta', x')\ ) \end{gathered}$$

hence by (2) the extension $(\mathtt{g}', \mathtt{g}'_{n+1}; \mathtt{f}', \mathtt{f}'_{m+1})$ of $\langle\ \mathtt{g}' \mid \mathtt{f}'\ \rangle$ via $\langle\ g' \mid f'\ \rangle$ (Lemma 3.2) is a canonical context homotopy equivalence $\langle\ \mathtt{d}' \mid \mathtt{c}'\ \rangle$ between $\delta', \underline{x} \equiv \delta$ and $\gamma', x \equiv \gamma$ and:

$$\lfloor \delta', \underline{x} \rfloor\ (\alpha(\delta'), \alpha_1(\delta', \underline{x})) : (\mathtt{g}, \mathtt{g}_{n+1})(\delta', \underline{x}) = (\mathtt{g}', \mathtt{g}'_{n+1})(\delta', \underline{x})$$

that is:

$$\lfloor \delta \rfloor\ \mathtt{d}(\delta) \equiv (\mathtt{g}, \mathtt{g}_{n+1})(\delta) = (\mathtt{g}', \mathtt{g}'_{n+1})(\delta) \equiv \mathtt{d}'(\delta)$$

hence we are done. □

Lemma 4.5. *Suppose that we are given a canonical homotopy equivalence $(\mathtt{f}; \mathtt{g})$ between $\gamma : \Gamma$ and $\delta : \Delta$ and suppose that there is a canonical homotopy equivalence $(\mathtt{g}'; \mathtt{f}')$ between δ and γ such that $\lfloor \gamma \rfloor\ \mathtt{f}(\gamma) = \mathtt{f}'(\gamma)$ and equivalently $\lfloor \delta \rfloor\ \alpha(\delta) : \mathtt{g}(\delta) = \mathtt{g}'(\delta)$.*

If we are given judgements:

$$\lfloor \gamma : \Gamma \rfloor\ S(\gamma) : \textsc{Type} \quad \textit{and} \quad \lfloor \delta : \Delta \rfloor\ T(\delta) : \textsc{Type}$$

and if there is a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$:

$$\begin{gathered} \lfloor \gamma : \Gamma, x : S(\gamma) \rfloor\ \varphi(\gamma, x) : T(\mathtt{f}(\gamma)) \\ \lfloor \gamma : \Gamma, \underline{x}' : T(\mathtt{f}(\gamma)) \rfloor\ \psi(\gamma, \underline{x}') : S(\gamma) \end{gathered}$$

between $S(\gamma)$ and $T(\delta)$ relative to a context homotopy equivalence $(\mathtt{f}; \mathtt{g})$, then there is a canonical homotopy equivalence $\langle\ \psi' \mid \varphi'\ \rangle$:

$$\begin{gathered} \lfloor \delta : \Delta, \underline{x} : T(\delta) \rfloor\ \psi'(\delta, \underline{x}) : S(\mathtt{g}'(\delta)) \\ \lfloor \delta : \Delta, x' : S(\mathtt{g}'(\delta)) \rfloor\ \varphi'(\delta, x') : T(\delta) \end{gathered}$$

relative to $(\mathtt{g}'; \mathtt{f}')$ such that:

$$\begin{gathered} \lfloor \delta : \Delta, \underline{x} : T(\delta) \rfloor\ \psi(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x}) = \alpha(\delta)^* \psi'(\delta, \underline{x}) \\ \textit{and equivalently } \lfloor \delta : \Delta, x' : S(\mathtt{g}'(\delta)) \rfloor\ \varphi(\mathtt{g}(\delta), \alpha(\delta)^* x') = \mathtt{q}(\delta)^* \varphi'(\delta, x'). \end{gathered}$$

Proof. By induction on the complexity of $\langle\ \varphi \mid \psi\ \rangle$.

If $\langle\ \varphi \mid \psi\ \rangle$ is of the form (a) then we are done.

If $\langle\ \varphi \mid \psi\ \rangle$ is of the form (b) then $S(\gamma) \equiv \Pi_{x:A(\gamma)}B(\gamma,x)$ and $T(\delta) \equiv \Pi_{\underline{x}:A'(\delta)}B'(\delta,\underline{x})$ for some judgements:

- $\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ B(\gamma,x) : \textsc{Type}$
- $\lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ B'(\delta,\underline{x}) : \textsc{Type}$

and there are a canonical homotopy equivalence $\langle\ f_1 \mid g_1\ \rangle$:

$$\begin{gathered}\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma,x) : A'(\mathtt{f}(\gamma)) \\ \lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma,\underline{x}') : A(\gamma)\end{gathered}$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$ and a canonical homotopy equivalence $\langle\ f_2 \mid g_2\ \rangle$:

$$\begin{gathered}\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma,x) \rfloor\ f_2(\gamma,x,y) : B'(\mathtt{f}(\gamma), f_1(\gamma,x)) \\ \lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma,x)) \rfloor\ g_2(\gamma,x,\underline{y}') : B(\gamma,x)\end{gathered}$$

between $B(\gamma,x)$ and $B'(\delta,\underline{x})$ relative to the extension $(\mathtt{f},\mathtt{f}_{m+1};\mathtt{g},\mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1;g_1)$ (Lemma 3.2), in such a way that the homotopy equivalence $(f^\Pi;g^\Pi)$ of Lemma 3.3 is the given $\langle\ \varphi \mid \psi\ \rangle$. By inductive hypothesis there are a canonical homotopy equivalence $\langle\ g_1' \mid f_1'\ \rangle$:

$$\begin{gathered}\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ g_1'(\delta,\underline{x}) : A(\mathtt{g}'(\delta)) \\ \lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta)) \rfloor\ f_1'(\delta,x') : A'(\delta)\end{gathered}$$

between $A'(\delta)$ and $A(\gamma)$ relative to $(\mathtt{g}';\mathtt{f}')$ and a canonical homotopy equivalence $\langle\ g_2' \mid f_2'\ \rangle$:

$$\begin{gathered}\lfloor \delta : \Delta, \underline{x} : A'(\delta), \underline{y} : B'(\delta,x) \rfloor\ g_2'(\delta,\underline{x},\underline{y}) : B(\mathtt{g}'(\delta), g_1'(\delta,\underline{x})) \\ \lfloor \delta : \Delta, \underline{x} : A'(\delta), y' : B(\mathtt{g}'(\delta), g_1'(\delta,\underline{x})) \rfloor\ f_2'(\delta,\underline{x},y') : B'(\delta,\underline{x})\end{gathered}$$

between $B'(\delta,\underline{x})$ and $B(\gamma,x)$ relative to the extension $(\mathtt{g}',\mathtt{g}'_{n+1};\mathtt{f}',\mathtt{f}'_{m+1})$ of $(\mathtt{g}';\mathtt{f}')$ via $(g_1';f_1')$, in such a way that:

$$\begin{gathered}\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \alpha_1(\delta,\underline{x}) : g_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x}) = \alpha(\delta)^* g_1'(\delta,\underline{x}) \\ \text{i.e. } \lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta)) \rfloor\ f_1(\mathtt{g}(\delta), \alpha(\delta)^* x') = \mathtt{q}(\delta)^* f_1'(\delta,x')\end{gathered}$$

and that:

$$\begin{gathered}\lfloor \delta : \Delta, \underline{x} : A'(\delta), \underline{y} : B'(\delta,\underline{x}) \rfloor\ g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta,\underline{x}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta,\underline{x}))^*\underline{y}) = \\ (\alpha(\delta), \alpha_1(\delta,\underline{x}))^* g_1'(\delta,\underline{x},\underline{y}) \\ \text{i.e. } \lfloor \delta : \Delta, \underline{x} : A'(\delta), y' : B(\mathtt{g}'(\delta), g_1'(\delta,\underline{x})) \rfloor\ f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta,\underline{x}), (\alpha(\delta), \alpha_1(\delta,\underline{x}))^* y') = \\ (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta,\underline{x}))^* f_2'(\delta,\underline{x},y')\end{gathered}$$

where we remind that:

$$\mathtt{g}_{n+1}(\delta,\underline{x}) \equiv g_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x})$$

$$\mathtt{q}_{n+1}(\delta,\underline{x}) \equiv q_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x})^{-1}$$

in context $\delta,\underline{x}$, being $\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ q_1(\gamma,\underline{x}') : \underline{x}' = f_1(\gamma, g_1(\gamma,\underline{x}'))$. Then $\langle\ g'^\Pi \mid f'^\Pi\ \rangle$ is a canonical homotopy equivalence between $\Pi_{\underline{x}:A'(\delta)}B'(\delta,\underline{x})$ and $\Pi_{x:A(\gamma)}B(\gamma,x)$ and relative to $(\mathtt{g}';\mathtt{f}')$, by (b). We are left to verify that:

$$\lfloor \delta, z' : \Pi_{x':A(g'(\delta))}B(\mathtt{g}'(\delta),x') \rfloor\ f^\Pi(\mathtt{g}(\delta), \alpha(\delta)^* z') = \mathtt{q}(\delta)^* f'^\Pi(\delta,z').$$

Let us observe that $f'^{\Pi}(\delta, z') \equiv$

$$\begin{aligned}
&\equiv \lambda \underline{x} : A'(\delta).f'_2(\delta, \underline{x}, \mathsf{ev}(z', g'_1(\delta, \underline{x}))) \\
&= \lambda \underline{x}.(\mathtt{q}(\delta)^{-1})^* q_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x})^* f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), (\alpha(\delta), \alpha_1(\delta, \underline{x}))^* \mathsf{ev}(z', g'_1(\delta, \underline{x}))) \\
&= \lambda \underline{x}.(\mathtt{q}(\delta)^{-1})^* q_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x})^* f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), \mathsf{ev}(\alpha(\delta)^* z', g_1(\mathtt{g}(\delta), q(\delta)^* \underline{x})))
\end{aligned}$$

in context δ, z', where the first propositional equality follows by:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta), y' : B(\mathtt{g}'(\delta), g'_1(\delta, \underline{x}))\rfloor$$
$$f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), (\alpha(\delta), \alpha_1(\delta, \underline{x}))^* y') = (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \underline{x}))^* f'_2(\delta, \underline{x}, y')$$

and the second by:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta)\rfloor \ \alpha_1(\delta, \underline{x}) : g_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x}) = \alpha(\delta)^* g'_1(\delta, \underline{x}).$$

Therefore $q(\delta)^* f'^{\Pi}(\delta, z') =$

$$\begin{aligned}
&= q(\delta)^* \lambda \underline{x}.(\mathtt{q}(\delta)^{-1})^* q_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x})^* f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), \mathsf{ev}(\alpha(\delta)^* z', \mathtt{g}_{n+1}(\delta, \underline{x}))) \\
&= \lambda \underline{x}'' : A'(\mathtt{f}(\mathtt{g}(\delta))).q_1(\mathtt{g}(\delta), \underline{x}'')^* f_2(\mathtt{g}(\delta), g_1(\mathtt{g}(\delta), \underline{x}''), \mathsf{ev}(\alpha(\delta)^* z', g_1(\mathtt{g}(\delta), \underline{x}''))) \\
&\equiv f^{\Pi}(\mathtt{g}(\delta), \alpha(\delta)^* z')
\end{aligned}$$

in context δ, z', where the first propositional equality follows by propositional function extensionality and propositional functoriality and the second by multiple (generalised) path induction on $q(\delta)$. We are done.

If $\langle\ \varphi \mid \psi\ \rangle$ is of the form (c) then as before $S(\gamma) \equiv \Sigma_{x:A(\gamma)} B(\gamma, x)$ and $T(\delta) \equiv \Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ for some judgements:

- $\lfloor \gamma : \Gamma\rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma : \Gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \textsc{Type}$
- $\lfloor \delta : \Delta\rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor \delta : \Delta, \underline{x} : A'(\delta)\rfloor\ B'(\delta, \underline{x}) : \textsc{Type}$

and there are a canonical homotopy equivalence $\langle\ f_1 \mid g_1\ \rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma)\rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma))\rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and a canonical homotopy equivalence $\langle\ f_2 \mid g_2\ \rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x)\rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x))\rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f}; \mathtt{g})$ via $(f_1; g_1)$ (Lemma 3.2), in such a way that the homotopy equivalence $(f^{\Sigma}; g^{\Sigma})$ of Lemma 3.4 is the given $\langle\ \varphi \mid \psi\ \rangle$. Again, by inductive hypothesis there are a canonical homotopy equivalence $\langle\ g'_1 \mid f'_1\ \rangle$:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta)\rfloor\ g'_1(\delta, \underline{x}) : A(\mathtt{g}'(\delta))$$
$$\lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta))\rfloor\ f'_1(\delta, x') : A'(\delta)$$

between $A'(\delta)$ and $A(\gamma)$ relative to $(\mathtt{g}'; \mathtt{f}')$ and a canonical homotopy equivalence $\langle\ g'_2 \mid f'_2\ \rangle$:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta), \underline{y} : B'(\delta, x)\rfloor\ g'_2(\delta, \underline{x}, \underline{y}) : B(\mathtt{g}'(\delta), g'_1(\delta, \underline{x}))$$
$$\lfloor \delta : \Delta, \underline{x} : A'(\delta), y' : B(\mathtt{g}'(\delta), g'_1(\delta, \underline{x}))\rfloor\ f'_2(\delta, \underline{x}, y') : B'(\delta, \underline{x})$$

between $B'(\delta, \underline{x})$ and $B(\gamma, x)$ relative to the extension $(\mathtt{g}', \mathtt{g}'_{n+1}; \mathtt{f}', \mathtt{f}'_{m+1})$ of $(\mathtt{g}'; \mathtt{f}')$ via $(g'_1; f'_1)$, in such a way that:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor \ \alpha_1(\delta, \underline{x}) : g_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x}) = \alpha(\delta)^* g_1'(\delta, \underline{x})$$
$$\text{i.e. } \lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta)) \rfloor \ f_1(\mathtt{g}(\delta), \alpha(\delta)^* x') = \mathtt{q}(\delta)^* f_1'(\delta, x')$$

and that:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta), \underline{y} : B'(\delta, \underline{x}) \rfloor \ g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \underline{x}))^* \underline{y}) = (\alpha(\delta), \alpha_1(\delta, \underline{x}))^* g_1'(\delta, \underline{x}, \underline{y})$$
$$\text{i.e. } \lfloor \delta : \Delta, \underline{x} : A'(\delta), y' : B(\mathtt{g}'(\delta), g_1'(\delta, \underline{x})) \rfloor \ f_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), (\alpha(\delta), \alpha_1(\delta, \underline{x}))^* y') = (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \underline{x}))^* f_2'(\delta, \underline{x}, y')$$

where, as before, we remind that:

$$\mathtt{g}_{n+1}(\delta, \underline{x}) \equiv g_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x})$$
$$\mathtt{q}_{n+1}(\delta, \underline{x}) \equiv q_1(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{x})^{-1}$$

in context $\delta, \underline{x}$, being $\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor \ q_1(\gamma, \underline{x}') : \underline{x}' = f_1(\gamma, g_1(\gamma, \underline{x}'))$. Then $\langle \ g'^{\Sigma} \mid f'^{\Sigma} \ \rangle$ is a canonical homotopy equivalence between $\Sigma_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})$ and $\Sigma_{x:A(\gamma)} B(\gamma, x)$ and relative to $(\mathtt{g}'; \mathtt{f}')$, by `(c)`. We are left to verify that:

$$\lfloor \delta, \underline{u} : \Sigma_{x':A'(\delta)} B'(\delta, x') \rfloor \ g^{\Sigma}(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{u}) = \alpha(\delta)^* g'^{\Sigma}(\delta, \underline{u}).$$

Let us observe that:

$$\begin{aligned}
\lfloor \delta, \underline{u} \rfloor \ \alpha_1(\delta, \pi_1 \underline{u}) : \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}) &= \alpha(\delta)^* g_1'(\delta, \pi_1 \underline{u}) \\
\lfloor \delta, \underline{u} \rfloor \ g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \pi_1 \underline{u}))^* \pi_2 \underline{u}) &= (\alpha(\delta), \alpha_1(\delta, \pi_1 \underline{u}))^* g_2'(\delta, \pi_1 \underline{u}, \pi_2 \underline{u}) \\
&= \alpha_1(\delta, \pi_1 \underline{u})^* \alpha(\delta)^* g_2'(\delta, \pi_1 \underline{u}, \pi_2 \underline{u})
\end{aligned}$$

where the first equality in the second judgement follows by:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta), \underline{y} : B'(\delta, \underline{x}) \rfloor$$
$$g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \underline{x}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \underline{x}))^* \underline{y}) = (\alpha(\delta), \alpha_1(\delta, \underline{x}))^* g_1'(\delta, \underline{x}, \underline{y})$$

and the second by (generalised) path induction on $\alpha_1(\delta, \pi_1 \underline{u})$. Therefore:

$$\begin{aligned}
&\langle \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}), g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \pi_1 \underline{u}))^* \pi_2 \underline{u}) \rangle \\
&\quad = \langle \alpha(\delta)^* g_1'(\delta, \pi_1 \underline{u}), \alpha(\delta)^* g_2'(\delta, \pi_1 \underline{u}, \pi_2 \underline{u}) \rangle \\
&\quad = \alpha(\delta)^* \langle g_1'(\delta, \pi_1 \underline{u}), g_2'(\delta, \pi_1 \underline{u}, \pi_2 \underline{u}) \rangle \\
&\quad \equiv \alpha(\delta)^* g'^{\Sigma}(\delta, \underline{u})
\end{aligned}$$

in context $\delta, \underline{u}$, where the second propositional equality follows by multiple (generalised) path induction on $\alpha(\delta)$. Finally $g^{\Sigma}(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{u}) \equiv$

$$\begin{aligned}
&\equiv \langle g_1(\mathtt{g}(\delta), \pi_1 \mathtt{q}(\delta)^* \underline{u}), g_2(\mathtt{g}(\delta), g_1(\mathtt{g}(\delta), \pi_1 \mathtt{q}(\delta)^* \underline{u}), q_1(\mathtt{g}(\delta), \pi_1 \mathtt{q}(\delta)^* \underline{u})^* \pi_2 \mathtt{q}(\delta)^* \underline{u}) \rangle \\
&= \langle \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}), g_2(\mathtt{g}(\delta), \mathtt{g}_{n+1}(\delta, \pi_1 \underline{u}), (\mathtt{q}(\delta), \mathtt{q}_{n+1}(\delta, \pi_1 \underline{u}))^* \pi_2 \underline{u}) \rangle
\end{aligned}$$

in context $\delta, \underline{u}$, where the propositional equality follows by multiple (generalised) path induction on $\mathtt{q}(\delta)$ followed by (generalised) path induction on $q_1(\mathtt{g}(\delta), \pi_1 \mathtt{q}(\delta)^* \underline{u})$. We are done.

Notation for the last section of the proof. Whenever:

$$\lfloor \omega : \Omega \rfloor \ x(\omega), y(\omega) : O(\omega)$$
$$\lfloor \omega \rfloor \ \alpha : x(\omega) = y(\omega)$$
$$\lfloor \omega, \omega' : \Omega \rfloor \ p : \omega' = \omega$$

we denote as $p^*\alpha : x(\omega') = y(\omega')$ the usual transport of the term $\alpha(\omega)$ from the type $x(\omega) = y(\omega)$ to the type $x(\omega') = y(\omega')$. However, the operation p^* is defined on the terms of $O(\omega)$ as well, hence the terms $p^*x(\omega), p^*y(\omega) : O(\omega')$ are defined. Therefore the operation p^* might be extended as usual to a propositional functor from $O(\omega)$ to $O(\omega')$ and a term $p^*\alpha : p^*x(\omega) = p^*y(\omega)$ is defined. In the last section, in order to distinguish between the terms:

$$p^*\alpha : x(\omega') = y(\omega') \text{ and } p^*\alpha : p^*x(\omega) = p^*y(\omega)$$

that we usually indicate by the same notation, we indicate the *latter* as $\underline{p^*\alpha}$. Observe that:

$$\lfloor p : \omega' = \omega \rfloor \quad \begin{array}{ccc} x(\omega') & \xRightarrow{\;p^*\alpha\;} & y(\omega') \\ {\scriptstyle x(p)}\Big\Downarrow & /\!\!/ & \Big\Downarrow{\scriptstyle y(p)} \\ p^*x(\omega) & \xRightarrow[\underline{p^*\alpha}]{} & p^*y(\omega). \end{array}$$

In order to verify this, since $\mathsf{r}(\omega')^*\alpha = \alpha$ and since $x(\mathsf{r}(\omega'))$ and $y(\mathsf{r}(\omega'))$ are propositionally equal to the canonical proofs $\xi(x(\omega'))$ and $\xi(y(\omega'))$ that $x(\omega') = \mathsf{r}(\omega')^*x(\omega')$ and $y(\omega') = \mathsf{r}(\omega')^*y(\omega')$ and by path induction, it is enough to verify that:

$$\lfloor \omega \rfloor \quad \begin{array}{ccc} x(\omega') & \xRightarrow{\;\alpha\;} & y(\omega') \\ {\scriptstyle \xi(x(\omega'))}\Big\Downarrow & /\!\!/ & \Big\Downarrow{\scriptstyle \xi(y(\omega'))} \\ \mathsf{r}(\omega')^*x(\omega') & \xRightarrow[\underline{\mathsf{r}(\omega')^*\alpha}]{} & \mathsf{r}(\omega')^*y(\omega'). \end{array}$$

which is true because $\xi(-)$ is a homotopy.

If $\langle\, \varphi \mid \psi \,\rangle$ is of the form `(d)` then $S(\gamma) \equiv s_1(\gamma) = s_2(\gamma)$ and $T(\delta) \equiv t_1(\delta) = t_2(\delta)$ for some judgements:

$$\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \quad \text{and} \quad \lfloor \delta : \Delta \rfloor\ A'(\delta) : \textsc{Type}$$

and some judgements:

$$\lfloor \gamma : \Gamma \rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma) \quad \text{and} \quad \lfloor \delta : \Delta \rfloor\ t_1(\delta), t_2(\delta) : A'(\delta)$$

and there are a canonical homotopy equivalence $\langle\, f \mid g \,\rangle$:

$$\begin{array}{c} \lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f(\gamma, x) : A'(\mathtt{f}(\gamma)) \\ \lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g(\gamma, \underline{x}') : A(\gamma) \end{array}$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and judgements:

$$\begin{array}{c} \lfloor \gamma : \Gamma \rfloor\ r_1(\gamma) : f(\gamma, s_1(\gamma)) = t_1(\mathtt{f}(\gamma)) \\ \lfloor \gamma : \Gamma \rfloor\ r_2(\gamma) : f(\gamma, s_2(\gamma)) = t_2(\mathtt{f}(\gamma)). \end{array}$$

in such a way that the homotopy equivalence $(f^=, g^=)$ of Lemma 3.5 is the given $\langle\, \varphi \mid \psi \,\rangle$. By inductive hypothesis there is a canonical homotopy equivalence $\langle\, g' \mid f' \,\rangle$:

$$\begin{array}{c} \lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ g'(\delta, \underline{x}) : A(\mathtt{g}'(\delta)) \\ \lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta)) \rfloor\ f'(\delta, x') : A'(\delta) \end{array}$$

between $A'(\delta)$ and $A(\gamma)$ relative to $(\mathtt{g}'; \mathtt{f}')$ such that:

$$\lfloor \delta : \Delta, \underline{x} : A'(\delta) \rfloor\ \alpha_1(\delta, \underline{x}) : g(\mathtt{g}(\delta), \mathtt{q}(\delta)^*\underline{x}) = \alpha(\delta)^* g'(\delta, \underline{x})$$
$$\text{i.e. } \lfloor \delta : \Delta, x' : A(\mathtt{g}'(\delta)) \rfloor\ f(\mathtt{g}(\delta), \alpha(\delta)^* x') = \mathtt{q}(\delta)^* f'(\delta, x').$$

For $i \in \{1, 2\}$, let us observe that in context δ:

$$\alpha_1(\delta, t_i(\delta))^{-1} : \alpha(\delta)^* g'(\delta, t_i(\delta)) = g(\mathtt{g}(\delta), \mathtt{q}(\delta)^* t_i(\delta))$$
$$g(\mathtt{g}(\delta), t_i(\mathtt{q}(\delta))^{-1}) : g(\mathtt{g}(\delta), \mathtt{q}(\delta)^* t_i(\delta)) = g(\mathtt{g}(\delta), t_i(\mathtt{f}(\mathtt{g}(\delta))))$$
$$g(\mathtt{g}(\delta), r_i(\mathtt{g}(\delta))^{-1}) : g(\mathtt{g}(\delta), t_i(\mathtt{f}(\mathtt{g}(\delta)))) = g(\mathtt{g}(\delta), f(\mathtt{g}(\delta), s_i(\mathtt{g}(\delta))))$$
$$p(\mathtt{g}(\delta), s_i(\mathtt{g}(\delta))) : g(\mathtt{g}(\delta), f(\mathtt{g}(\delta), s_i(\mathtt{g}(\delta)))) = s_i(\mathtt{g}(\delta))$$
$$s_i(\alpha(\gamma)) : s_i(\mathtt{g}(\delta)) = \alpha(\delta)^* s_i(\mathtt{g}'(\delta))$$

and let us call $a_i(\delta) : \alpha(\delta)^* g'(\delta, t_i(\delta)) = \alpha(\delta)^* s_i(\mathtt{g}'(\delta))$. Let $r'_i(\delta) : g'(\delta, t_i(\delta)) = s_i(\mathtt{g}'(\delta))$ be such that:

$$\underline{\alpha(\delta)^* r'_i(\delta)} = a_i(\delta).$$

Hence $\langle\, g'^{=} \mid f'^{=} \,\rangle$ is a canonical homotopy equivalence between $t_1(\delta) = t_2(\delta)$ and $s_1(\gamma) = s_2(\gamma)$ relative to $(\mathtt{g}'; \mathtt{f}')$, by `(d)`, and we are left to verify that:

$$\lfloor \delta : \Delta, \underline{p} : t_1(\delta) = t_2(\delta) \rfloor\ g^{=}(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{p}) = \alpha(\delta)^* g'^{=}(\delta, \underline{p})$$

that is in fact true, as $\lfloor \delta, \underline{p} \rfloor\ \alpha(\delta)^* g'^{=}(\delta, \underline{p}) \equiv$

$$\begin{aligned}
&\equiv \alpha(\delta)^*(r'_1(\delta)^{-1} \bullet g'(\delta, \underline{p}) \bullet r'_2(\delta)) \\
&= s_1(\alpha(\delta)) \bullet \underline{\alpha(\delta)^*(r'_1(\delta)^{-1} \bullet g'(\delta, \underline{p}) \bullet r'_2(\delta))} \bullet s_2(\alpha(\delta))^{-1} \\
&= s_1(\alpha(\delta)) \bullet \underline{\alpha(\delta)^* r'_1(\delta)^{-1}} \bullet \underline{\alpha(\delta)^* g'(\delta, \underline{p})} \bullet \underline{\alpha(\delta)^* r'_2(\delta)} \bullet s_2(\alpha(\delta))^{-1} \\
&= s_1(\alpha(\delta)) \bullet a_1(\gamma)^{-1} \bullet \underline{\alpha(\delta)^* g'(\delta, \underline{p})} \bullet a_2(\gamma) \bullet s_2(\alpha(\delta))^{-1} \\
&= p(\mathtt{g}(\delta), s_1(\mathtt{g}(\delta)))^{-1} \bullet g(\mathtt{g}(\delta), r_1(\mathtt{g}(\delta))) \bullet g(\mathtt{g}(\delta), t_1(\mathtt{q}(\delta))) \bullet \alpha_1(\delta, t_1(\delta)) \bullet \underline{\alpha(\delta)^* g'(\delta, \underline{p})} \bullet \\
&\quad \alpha_1(\delta, t_2(\delta))^{-1} \bullet g(\mathtt{g}(\delta), t_2(\mathtt{q}(\delta))^{-1}) \bullet g(\mathtt{g}(\delta), r_2(\mathtt{g}(\delta))^{-1}) \bullet p(\mathtt{g}(\delta), s_2(\mathtt{g}(\delta))) \\
&= p(\mathtt{g}(\delta), s_1(\mathtt{g}(\delta)))^{-1} \bullet g(\mathtt{g}(\delta), r_1(\mathtt{g}(\delta))) \bullet g(\mathtt{g}(\delta), t_1(\mathtt{q}(\delta))) \bullet g(\mathtt{g}(\delta), \underline{\mathtt{q}(\delta)^* \underline{p}}) \bullet \\
&\quad g(\mathtt{g}(\delta), t_2(\mathtt{q}(\delta))^{-1}) \bullet g(\mathtt{g}(\delta), r_2(\mathtt{g}(\delta))^{-1}) \bullet p(\mathtt{g}(\delta), s_2(\mathtt{g}(\delta))) \\
&= p(\mathtt{g}(\delta), s_1(\mathtt{g}(\delta)))^{-1} \bullet g(\mathtt{g}(\delta), r_1(\mathtt{g}(\delta))) \bullet g(\mathtt{g}(\delta), t_1(\mathtt{q}(\delta)) \bullet \underline{\mathtt{q}(\delta)^* \underline{p}} \bullet t_2(\mathtt{q}(\delta))^{-1}) \bullet \\
&\quad g(\mathtt{g}(\delta), r_2(\mathtt{g}(\delta))^{-1}) \bullet p(\mathtt{g}(\delta), s_2(\mathtt{g}(\delta))) \\
&= p(\mathtt{g}(\delta), s_1(\mathtt{g}(\delta)))^{-1} \bullet g(\mathtt{g}(\delta), r_1(\mathtt{g}(\delta))) \bullet g(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{p}) \bullet \\
&\quad g(\mathtt{g}(\delta), r_2(\mathtt{g}(\delta))^{-1}) \bullet p(\mathtt{g}(\delta), s_2(\mathtt{g}(\delta))) \\
&= p(\mathtt{g}(\delta), s_1(\mathtt{g}(\delta)))^{-1} \bullet g(\mathtt{g}(\delta), r_1(\mathtt{g}(\delta)) \bullet \mathtt{q}(\delta)^* \underline{p} \bullet r_2(\mathtt{g}(\delta))^{-1}) \bullet p(\mathtt{g}(\delta), s_2(\mathtt{g}(\delta))) \\
&\equiv g^{=}(\mathtt{g}(\delta), \mathtt{q}(\delta)^* \underline{p})
\end{aligned}$$

hence we are done. □

Lemma 4.6. *Let $\gamma : \Gamma$, $\delta : \Delta$ and $\omega : \Omega$ be contexts, let $(\mathtt{f}; \mathtt{g})$ be a canonical context homotopy equivalence between γ and δ and let $(\mathtt{f}', \mathtt{g}')$ be a context homotopy equivalence between δ and ω.*

If we are given judgements $\lfloor \gamma \rfloor\ S(\gamma) : \mathrm{Type}$, $\lfloor \delta \rfloor\ T(\delta) : \mathrm{Type}$ and $\lfloor \omega \rfloor\ U(\omega) : \mathrm{Type}$ and canonical homotopy equivalences $\langle\, \varphi \mid \psi \,\rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and $\langle\, \varphi' \mid \psi' \,\rangle$ between $T(\delta)$ and $U(\omega)$ relative to $(\mathtt{f}'; \mathtt{g}')$, then there exists a canonical homotopy equivalence $\langle\, \varphi'' \mid \psi'' \,\rangle$ between $S(\gamma)$ and $U(\omega)$ relative to $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ and such that:

$$\lfloor \gamma, s : S(\gamma) \rfloor \; \varphi''(\gamma, s) = \varphi'(\mathtt{f}(\gamma), \varphi(\gamma, s))$$
$$\textit{and equivalently } \lfloor \gamma, u'' : U(\mathtt{f}'(\mathtt{f}(\gamma))) \rfloor \; \psi''(\gamma, u'') = \psi(\gamma, \psi'(\mathtt{f}(\gamma), u'')).$$

Proof. By induction on the complexity of $\langle \varphi \mid \psi \rangle$. □

Lemma 4.7. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be contexts and let $(\mathtt{f};\mathtt{g})$ and $(\mathtt{f}';\mathtt{g}')$ be context homotopy equivalences between γ and δ. Let us assume that $\lfloor \gamma \rfloor \; \alpha(\gamma) : \mathtt{f}(\gamma) = \mathtt{f}'(\gamma)$ and equivalently $\lfloor \delta \rfloor \; \mathtt{g}(\delta) = \mathtt{g}'(\delta)$.*

If we are given judgements $\lfloor \gamma \rfloor \; S(\gamma) : \mathrm{TYPE}$ and $\lfloor \delta \rfloor \; T(\delta) : \mathrm{TYPE}$ and a canonical homotopy equivalence $\langle \varphi \mid \psi \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f};\mathtt{g})$, then there is a canonical homotopy equivalence $\langle \varphi' \mid \psi' \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}';\mathtt{g}')$ such that:

$$\lfloor \gamma, s : S(\gamma) \rfloor \; \varphi(\gamma, s) = \alpha(\gamma)^* \varphi'(\gamma, s)$$
$$\textit{and equivalently } \lfloor \gamma, t' : T(\mathtt{f}'(\gamma)) \rfloor \; \psi(\gamma, \alpha(\gamma)^* t') = \psi'(\gamma, t').$$

Proof. By induction on the complexity of $\langle \varphi \mid \psi \rangle$. □

Proposition 4.8 (Transitivity)**.** *Let $\gamma : \Gamma$, $\delta : \Delta$ and $\omega : \Omega$ be contexts. If $\langle \mathtt{c} \mid \mathtt{d} \rangle$ is a canonical context homotopy equivalence between γ and δ and $\langle \mathtt{c}' \mid \mathtt{d}' \rangle$ is a canonical context homotopy equivalence between δ and ω then there is a canonical context homotopy equivalence $\langle \mathtt{c}'' \mid \mathtt{d}'' \rangle$ between γ and ω such that $\lfloor \gamma \rfloor \; \mathtt{c}''(\gamma) = \mathtt{c}'(\mathtt{c}(\gamma))$ and equivalently $\lfloor \delta \rfloor \; \mathtt{d}''(\delta) = \mathtt{d}(\mathtt{d}'(\delta))$.*

Proof. By induction on the complexity of $\langle \mathtt{c} \mid \mathtt{d} \rangle$. If $\langle \mathtt{c} \mid \mathtt{d} \rangle$ is of the form `(a)` then $\langle \mathtt{c}' \mid \mathtt{d}' \rangle$ is of the form `(a)` as well and we are done. Let $\langle \mathtt{c} \mid \mathtt{d} \rangle$ be of the form `(b)`, hence $\langle \mathtt{c}' \mid \mathtt{d}' \rangle$ is of the form `(b)` as well. Then $\gamma : \Gamma$, $\delta : \Delta$ and $\omega : \Omega$ are of the form $\gamma' : \Gamma', x : A(\gamma')$ and $\delta' : \Delta', \underline{x} : A'(\delta')$ and $\omega' : \Omega', \underline{\underline{x}} : A''(\Omega')$ respectively. Moreover there are a canonical context homotopy equivalence:

$$\lfloor \gamma' : \Gamma' \rfloor \; \mathtt{f}(\gamma') : \Delta'$$
$$\lfloor \delta' : \Delta' \rfloor \; \mathtt{g}(\delta') : \Gamma'$$

and a canonical homotopy equivalence:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor \; f(\gamma', x) : A'(\mathtt{f}(\gamma'))$$
$$\lfloor \gamma' : \Gamma', \underline{x}' : A'(\mathtt{f}(\gamma')) \rfloor \; g(\gamma', \underline{x}') : A(\gamma')$$

between $A(\gamma')$ and $A'(\delta')$ relative to $\langle \mathtt{f} \mid \mathtt{g} \rangle$ in such a way that the extension:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor \; (\mathtt{f}, \mathtt{f}_{m+1})(\gamma', x) : \Delta', A'$$
$$\lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor \; (\mathtt{g}, \mathtt{g}_{n+1})(\delta', \underline{x}) : \Gamma', A$$

of $\langle \mathtt{f} \mid \mathtt{g} \rangle$ via $\langle f \mid g \rangle$ (Lemma 3.2) is the given $\langle \mathtt{c} \mid \mathtt{d} \rangle$. Analogously, there are a canonical context homotopy equivalence:

$$\lfloor \delta' : \Delta' \rfloor \; \mathtt{f}'(\delta') : \Omega'$$
$$\lfloor \omega' : \Omega' \rfloor \; \mathtt{g}(\omega') : \Delta'$$

and a canonical homotopy equivalence:

$$\lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor \; f'(\delta', \underline{x}) : A''(\mathtt{f}'(\delta'))$$
$$\lfloor \delta' : \Delta', \underline{\underline{x}} : A''(\mathtt{f}'(\delta')) \rfloor \; g'(\delta', \underline{\underline{x}}') : A'(\delta')$$

between $A'(\delta')$ and $A''(\omega')$ relative to $\langle \mathtt{f}' \mid \mathtt{g}' \rangle$ in such a way that the extension:

$$\lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor \ (\mathtt{f}', \mathtt{f}'_{m+1})(\delta', \underline{x}) : \Omega', A''$$
$$\lfloor \omega' : \Omega', \underline{\underline{x}} : A''(\omega') \rfloor \ (\mathtt{g}', \mathtt{g}'_{n+1})(\omega', \underline{\underline{x}}) : \Delta', A'$$

of $\langle\, \mathtt{f}' \mid \mathtt{g}' \,\rangle$ via $\langle\, f' \mid g' \,\rangle$ (Lemma 3.2) is the given $\langle\, \mathtt{c}' \mid \mathtt{d}' \,\rangle$. By inductive hypothesis there is a canonical context homotopy equivalence $\langle\, \mathtt{f}'' \mid \mathtt{g}'' \,\rangle$ between γ' and ω' such that:

$$\lfloor \gamma' \rfloor \ \mathtt{f}''(\gamma') = \mathtt{f}'(\mathtt{f}(\gamma')) \text{ and } \lfloor \omega' \rfloor \ \mathtt{g}''(\omega') = \mathtt{g}(\mathtt{g}'(\omega'))$$

Let us consider the canonical homotopy equivalence:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor \ \tilde{f}(\gamma', x) \equiv f'(\mathtt{f}(\gamma'), f(\gamma', x)) : A''(\mathtt{f}'(\mathtt{f}(\gamma')))$$
$$\lfloor \gamma' : \Gamma', \underline{\underline{x}}'' : A''(\mathtt{f}'(\mathtt{f}(\gamma'))) \rfloor \ \tilde{g}(\gamma', \underline{\underline{x}}'') \equiv g(\gamma', g'(\mathtt{f}(\gamma'), \underline{\underline{x}}'')) : A(\gamma')$$

relative to $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ and let us observe that:

$$\begin{aligned}(\mathtt{f}'(\mathtt{f}(\gamma)), f'(\mathtt{f}(\gamma'), f(\gamma', x))) &\equiv (\mathtt{f}'(\mathtt{f}(\gamma)), \mathtt{f}'_{m+1}(\mathtt{f}(\gamma'), \mathtt{f}_{m+1}(\gamma', x))) \\ &\equiv (\mathtt{f}', \mathtt{f}'_{m+1})(\mathtt{f}, \mathtt{f}_{m+1})(\gamma', x) \\ &\equiv \mathtt{c}'(\mathtt{c}(\gamma))\end{aligned}$$

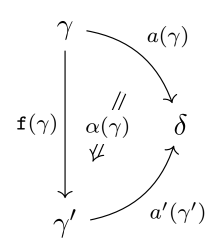
hence the extension of $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ via $(\tilde{f}; \tilde{g})$ is pairwise propositionally equal to $(\mathtt{c}'\mathtt{c}; \mathtt{d}\mathtt{d}')$. By Lemma 4.6, there is canonical homotopy equivalence $\langle\, \overline{f} \mid \overline{g} \,\rangle$ between $A(\gamma')$ and $A''(\omega')$ relative to $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ such that $\lfloor \gamma' \rfloor \ \overline{f}(\gamma') = \tilde{f}(\gamma')$ and equivalently $\lfloor \omega' \rfloor \ \overline{g}(\omega') = \tilde{g}(\omega')$. Then, again, the extension of $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ via $\langle\, \overline{f} \mid \overline{g} \,\rangle$ is pairwise propositionally equal to $(\mathtt{c}'\mathtt{c}; \mathtt{d}\mathtt{d}')$. Since $\lfloor \gamma' \rfloor \ \alpha(\gamma') : \mathtt{f}'(\mathtt{f}(\gamma')) = \mathtt{f}''(\gamma')$ and equivalently $\lfloor \omega' \rfloor \ \mathtt{g}(\mathtt{g}'(\omega')) = \mathtt{g}''(\omega')$, by Lemma 4.7 there is a canonical homotopy equivalence $\langle\, f'' \mid g'' \,\rangle$ between $A(\gamma')$ and $A''(\omega')$ relative to $\langle\, \mathtt{f}'' \mid \mathtt{g}'' \,\rangle$ and such that:

$$\lfloor \gamma', x : A(\gamma') \rfloor \ \overline{f}(\gamma', x) = \alpha(\gamma')^* f''(\gamma', x)$$
$$\text{and equivalently } \lfloor \gamma', \underline{\underline{x}}' : A''(\mathtt{f}''(\gamma')) \rfloor \ \overline{g}(\gamma', \alpha(\gamma')^* \underline{\underline{x}}') = g''(\gamma', \underline{\underline{x}}').$$

Therefore the extension $\langle\, \mathtt{f}'', \mathtt{f}''_{m+1} \mid \mathtt{g}'', \mathtt{g}''_{n+1} \,\rangle$ of $\langle\, \mathtt{f}'' \mid \mathtt{g}'' \,\rangle$ via $\langle\, f'' \mid g'' \,\rangle$ is a canonical context homotopy equivalence by (b) and $\langle\, \mathtt{f}'', \mathtt{f}''_{m+1} \mid \mathtt{g}'', \mathtt{g}''_{n+1} \,\rangle$ is pairwise propositionally equal to the extension of $(\mathtt{f}'\mathtt{f}; \mathtt{g}\mathtt{g}')$ via $\langle\, \overline{f} \mid \overline{g} \,\rangle$. In particular $\langle\, \mathtt{f}'', \mathtt{f}''_{m+1} \mid \mathtt{g}'', \mathtt{g}''_{n+1} \,\rangle$ is a canonical context homotopy equivalence between γ and δ that is pairwise propositionally equal to $(\mathtt{c}'\mathtt{c}; \mathtt{d}\mathtt{d}')$. □

We conclude the current subsection with a brief list of technical results that we are using in Section 5 and Section 6.

Lemma 4.9. *Let $\gamma : \Gamma$, $\gamma' : \Gamma'$ and $\delta : \Delta$ be contexts, let $(\mathtt{f}; \mathtt{g})$ be a context homotopy equivalence between γ and γ' and let $a(\gamma), a'(\gamma') : \Delta$ be context morphisms $\gamma \to \delta$ and $\gamma' \to \delta$ respectively such that:*



Then, whenever $\lfloor \delta \rfloor \ S(\delta) :$ Type, there is a canonical homotopy equivalence $\langle\, \varphi \mid \psi \,\rangle$ between $S(a(\gamma))$ and $S(a'(\gamma'))$ relative to $(\mathtt{f}; \mathtt{g})$ such that:

$$\lfloor \gamma, x : S(a(\gamma)) \rfloor\ \alpha(\gamma)^* \varphi(\gamma, x) = x \quad \textit{and equivalently}$$
$$\lfloor \gamma, \underline{x}' : S(a'(\mathtt{f}(\gamma))) \rfloor\ \psi(\gamma, \underline{x}') = \alpha(\gamma)^* \underline{x}'.$$

Proof. By induction on the complexity of $S(\delta)$. □

Corollary 4.10. *Let $\gamma : \Gamma$ and $\gamma' : \Gamma'$ be contexts and let $(\mathtt{f}; \mathtt{g})$ be a context homotopy equivalence between γ and γ'. Then, whenever $\lfloor \gamma' \rfloor\ S(\gamma') : \textsc{Type}$, there is a canonical homotopy equivalence $\langle\, \varphi \mid \psi \,\rangle$ between $S(\mathtt{f}(\gamma))$ and $S(\gamma')$ relative to $(\mathtt{f}; \mathtt{g})$ and such that:*

$$\lfloor \gamma, x : S(\mathtt{f}(\gamma)) \rfloor\ \varphi(\gamma, x) = \psi(\gamma, x) = x.$$

Proof. Follows by Lemma 4.9, where $\delta \equiv \gamma'$, $a(\gamma) \equiv \mathtt{f}(\gamma)$, $a'(\gamma') \equiv \gamma'$ and $\alpha(\gamma) \equiv \mathtt{r}(\mathtt{f}(\gamma))$. □

Lemma 4.11. *Let $\gamma : \Gamma$ and $\gamma' : \Gamma'$ be contexts and let $(\mathtt{f}; \mathtt{g})$ be a context homotopy equivalence between γ and γ'. Suppose that $\lfloor \gamma \rfloor\ S(\gamma) : \textsc{Type}$ and $\lfloor \gamma' \rfloor\ T(\gamma') : \textsc{Type}$ are canonically homotopy equivalent via some $\langle\, \varphi \mid \psi \,\rangle$ relative to $(\mathtt{f}; \mathtt{g})$. Let $\langle\, \mathtt{c} \mid \mathtt{d} \,\rangle$ be some canonical context homotopy equivalence between γ and itself such that $\langle\, \mathtt{c} \mid \mathtt{d} \,\rangle$ is pairwise propositionally equal to $(\lfloor \gamma \rfloor\ \gamma; \lfloor \gamma \rfloor\ \gamma)$ (whose existence is ensured by Proposition 4.2) and let $\lfloor \gamma \rfloor\ \alpha(\gamma) : \mathtt{c}(\gamma) = \gamma$.*

Then $S(\gamma)$ and $T(\mathtt{f}(\gamma))$ are canonically homotopy equivalent via some $\langle\, \varphi' \mid \psi' \,\rangle$ relative to $\langle\, \mathtt{c} \mid \mathtt{d} \,\rangle$ in such a way that:

$$\lfloor \gamma, x : S(\gamma) \rfloor\ \varphi'(\gamma, x) = \alpha(\gamma)^* \varphi(\gamma, x)$$

and equivalently $\lfloor \gamma, \underline{x}' : T(\mathtt{f}(\gamma)) \rfloor\ \psi'(\gamma, \alpha(\gamma)^* \underline{x}') = \psi(\gamma, \underline{x}')$.

Proof. By induction on the complexity of $\langle\, \varphi \mid \psi \,\rangle$. □

Lemma 4.12. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be contexts, let $(\mathtt{f}; \mathtt{g})$ be a canonical homotopy equivalence between δ and itself such that $\lfloor \delta \rfloor\ \alpha(\delta) : \mathtt{f}(\delta) = \delta$ and let $(\mathtt{f}'; \mathtt{g}')$ be a canonical homotopy equivalence between γ and itself such that $\lfloor \gamma \rfloor\ \alpha'(\gamma) : \gamma = \mathtt{f}'(\gamma)$. Moreover, let $\lfloor \gamma \rfloor\ a(\gamma) : \Delta$ be a context morphism and let $\lfloor \delta \rfloor\ S(\delta) : \textsc{Type}$ and $\lfloor \delta \rfloor\ T(\delta) : \textsc{Type}$ be such that there is a canonical homotopy equivalence $\langle\, \varphi \mid \psi \,\rangle$ between $S(\delta)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$. Then there is a canonical homotopy equivalence $\langle\, \varphi' \mid \psi' \,\rangle$ between $S(a(\gamma))$ and $T(a(\gamma))$ relative to $(\mathtt{f}'; \mathtt{g}')$ and such that:*

$$\lfloor \gamma, x : S(a(\gamma)) \rfloor\ \varphi(a(\gamma), x) = \alpha''(\gamma)^* \varphi'(\gamma, x)$$

i.e. $\lfloor \gamma, \underline{x} : T(a(\mathtt{f}'(\gamma))) \rfloor\ \psi(a(\gamma), \alpha''(\gamma)^* \underline{x}) = \psi'(\gamma, \underline{x})$ where $\alpha''(\gamma) \equiv \alpha(a(\gamma)) \bullet a(\alpha'(\gamma))$.

Proof. By induction on the complexity of $\langle\, \varphi \mid \psi \,\rangle$. □

4.2. **Properties of a restriction of the family.** In this subsection we restrict the family of the canonical context homotopy equivalences and prove additional properties satisfied by this restriction. Again, we remind that we use the notation $(\ \cdot\ ;\ \cdot\ )$ to indicate any homotopy equivalence between contexts or types in context, reserving the symbol $\langle \cdot \mid \cdot \rangle$ for the canonical ones. Moreover, we remind that a type judgement $\lfloor \delta \rfloor\ T(\delta) : \textsc{Type}$ is said to be an *h-proposition* if $\lfloor \delta, x_1, x_2 : T(\delta) \rfloor\ \alpha(\delta, x_1, x_2) : x_1 = x_2$. We start by giving the following inductive definition:

Definition 4.13. A type judgement $\lfloor \delta \rfloor\ T(\delta) : \textsc{Type}$ in some context $\delta : \Delta$ **has h-propositional identities** (or is **with h-propositional identities**) if it belongs to the smallest family $\mathcal{F}$ of type judgements that satisfies the following clauses:

- a judgement $\lfloor \gamma \rfloor\ S : \textsc{Type}$ (where S is a atomic type) belongs to $\mathcal{F}$;

- a judgement $\lfloor \gamma \rfloor\ \Pi_{x:A(\gamma)}B(\gamma,x)$: Type, for some $\lfloor \gamma \rfloor\ A(\gamma)$: Type of $\mathcal{F}$ and some $\lfloor \gamma, x : A(\gamma) \rfloor\ B(\gamma,x)$: Type of $\mathcal{F}$, belongs to $\mathcal{F}$;
- a judgement $\lfloor \gamma \rfloor\ \Sigma_{x:A(\gamma)}B(\gamma,x)$: Type, for some $\lfloor \gamma \rfloor\ A(\gamma)$: Type of $\mathcal{F}$ and some $\lfloor \gamma, x : A(\gamma) \rfloor\ B(\gamma,x)$: Type of $\mathcal{F}$, belongs to $\mathcal{F}$;
- an h-proposition $\lfloor \gamma \rfloor\ s_1(\gamma) = s_2(\gamma)$: Type, for some $\lfloor \gamma \rfloor\ A(\gamma)$: Type of $\mathcal{F}$ and some $\lfloor \gamma \rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma)$, belongs to $\mathcal{F}$.

Despite our choice of terminology, we would like to clarify that the notion of a type with h-propositional identities does not refer to the types T whose identity types $\lfloor x, y : T \rfloor\ x = y$: Type are h-propositions, i.e. the *h-sets*. Rather, it characterises types whose equalities *involved in their construction* are h-propositions.

Definition 4.14. Let $\gamma : \Gamma$ be a context $\gamma_1 : \Gamma_1, \gamma_2 : \Gamma_2(\gamma_1), ..., \gamma_n : \Gamma_n(\gamma_1, ..., \gamma_{n-1})$, where n might be 0. For any $i \in \{1, ..., n\}$, let γ^i be the context $\gamma_1, ..., \gamma_i$. We say that γ **has h-propositional identities** (or is **with h-propositional identities**) if, for every $i \in \{1, ..., n\}$, the judgement $\lfloor \gamma^i \rfloor\ \Gamma_i(\gamma^i)$ has h-propositional identities.

Lemma 4.15. *Let $\gamma : \Gamma$ be a context $\gamma_1 : \Gamma_1, \gamma_2 : \Gamma_2(\gamma_1), ..., \gamma_n : \Gamma_n(\gamma_1, ..., \gamma_{n-1})$ and let $\delta : \Delta$ be a context $\delta_1 : \Delta_1, \delta_2 : \Delta_2(\delta_1), ..., \delta_m : \Delta_m(\delta_1, ..., \delta_{m-1})$. If there is a canonical context homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between $\gamma : \Gamma$ and $\delta : \Delta$ then γ has h-propositional identities if and only if δ has h-propositional identities.*

Proof. By Proposition 4.4 we can assume w.l.o.g. that γ has h-propositional identities and we are left to verify that δ has h-propositional identities as well. By induction on the complexity of $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form `(1)` then we are done. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form `(2)` then $\gamma : \Gamma$ and $\delta : \Delta$ are of the form $\gamma' : \Gamma', x : A(\gamma')$ and $\delta' : \Delta', \underline{x} : A'(\delta')$ respectively and there are a canonical context homotopy equivalence $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$:

$$\lfloor \gamma' : \Gamma' \rfloor\ \mathtt{f}(\gamma') : \Delta'$$
$$\lfloor \delta' : \Delta' \rfloor\ \mathtt{g}(\delta') : \Gamma'$$

and a canonical homotopy equivalence $\langle\ f \mid g\ \rangle$:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor\ f(\gamma', x) : A'(\mathtt{f}(\gamma'))$$
$$\lfloor \gamma' : \Gamma', \underline{x}' : A'(\mathtt{f}(\gamma')) \rfloor\ g(\gamma', \underline{x}') : A(\gamma')$$

between $A(\gamma')$ and $A'(\delta')$ relative to $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$, in such a way that the extension:

$$\lfloor \gamma' : \Gamma', x : A(\gamma') \rfloor\ (\mathtt{f}, \mathtt{f}_{m+1})(\gamma', x) : \Delta', A'$$
$$\lfloor \delta' : \Delta', \underline{x} : A'(\delta') \rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})(\delta', \underline{x}) : \Gamma', A$$

of $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ via $\langle\ f \mid g\ \rangle$ (Lemma 3.2) is the canonical context homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$. By inductive hypothesis and being γ' a context with h-propositional identities, the context δ' has h-propositional identities. Moreover, by the following Lemma 4.16 and having $\lfloor \gamma' \rfloor\ A(\gamma')$: Type h-propositional identities, we deduce that $\lfloor \delta' \rfloor\ A'(\delta')$: Type has h-propositional identities. Therefore δ has h-propositional identities and we are done. □

Lemma 4.16. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be contexts and let $(\mathtt{f}; \mathtt{g})$ be a canonical context homotopy equivalence between γ and δ. Let $\lfloor \gamma : \Gamma \rfloor\ S(\gamma)$: Type and $\lfloor \gamma : \Delta \rfloor\ T(\delta)$: Type be such that there is a canonical homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between $S(\gamma)$ and $T(\delta)$ relative to $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$. Then $\lfloor \gamma : \Gamma \rfloor\ S(\gamma)$: Type has h-propositional identities if and only if $\lfloor \gamma : \Delta \rfloor\ T(\delta)$: Type has h-propositional identities.*

Proof. By induction on the complexity of $\langle\ \varphi \mid \psi\ \rangle$. If $\langle\ \varphi \mid \psi\ \rangle$ is of the form `(a)` then we are done. If $\langle\ \varphi \mid \psi\ \rangle$ is of the form `(b)` then $S(\gamma)$ is of the form $\Pi_{x:A(\gamma)}B(\gamma,x)$ and $T(\delta)$ is of the form $\Pi_{x':A'(\delta)}B(\delta,x')$, where $\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor\gamma,x\rfloor\ B(\gamma,x) : \textsc{Type}$ have h-propositional identities, $A(\gamma)$ is canonically homotopy equivalent to $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$ and $B(\gamma,x)$ is canonically homotopy equivalent to $B'(\delta,x')$ relative to the extension of $(\mathtt{f};\mathtt{g})$ via the given canonical homotopy equivalence between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$. By inductive hypothesis, the judgements $\lfloor\delta\rfloor\ A'(\delta) : \textsc{Type}$ and $\lfloor\delta,x'\rfloor\ B'(\delta,x') : \textsc{Type}$ have h-propositional identities, hence $T(\delta)$ has h-propositional identities and we are done. If $\langle\ \varphi \mid \psi\ \rangle$ is of the form `(c)` then we infer that $T(\delta)$ has h-propositional identities as for the case `(b)`. Finally, if $\langle\ \varphi \mid \psi\ \rangle$ is of the form `(d)` then $S(\gamma)$ is an h-proposition of the form $s_1(\gamma) = s_2(\gamma)$ for some $\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}$ with h-propositional identities and some $\lfloor\gamma\rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma)$, while $T(\delta)$ is of the form $t_1(\delta) = t_2(\delta)$ for some $\lfloor\delta\rfloor\ A'(\delta) : \textsc{Type}$ and some $\lfloor\delta\rfloor\ t_1(\delta), t_2(\delta) : A'(\delta)$, being $A(\gamma)$ and $A'(\delta)$ canonically homotopy equivalent relative to $(\mathtt{f};\mathtt{g})$ in such a way that the canonical homotopy equivalence propositionally identifies $s_i(\gamma)$ and $t_i(\gamma)$, for $i \in \{1,2\}$. By inductive hypothesis, the judgement $\lfloor\delta\rfloor\ A'(\delta) : \textsc{Type}$ has h-propositional identities. Moreover $t_1(\delta) = t_2(\delta)$ happens to be an h-proposition, since it is homotopy equivalent (via $\langle\ \varphi \mid \psi\ \rangle$) to an h-set. Hence $T(\delta)$ has h-propositional identities and we are done. □

Proposition 4.17. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be contexts with h-propositional identities. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ are canonical context homotopy equivalences between $\gamma : \Gamma$ and $\delta : \Delta$, then $\lfloor\gamma : \Gamma\rfloor\ \mathtt{c}(\gamma) = \mathtt{c}'(\gamma)$ (i.e. $\lfloor\delta : \Delta\rfloor\ \mathtt{d}(\delta) = \mathtt{d}'(\delta)$).*

Proof. By induction on the complexity of $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form `(1)` then $\gamma : \Gamma$ and $\delta : \Delta$ are the empty contexts hence $\mathtt{c} \equiv \mathtt{c}'$ and $\mathtt{d} \equiv \mathtt{d}'$. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is of the form `(2)` then $\gamma : \Gamma$ and $\delta : \Delta$ are of the form $\gamma' : \Gamma', x : A(\gamma')$ and $\delta' : \Delta', \underline{x} : A'(\delta')$ respectively and there are a canonical context homotopy equivalence $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$:

$$\lfloor\gamma' : \Gamma'\rfloor\ \mathtt{f}(\gamma') : \Delta'$$
$$\lfloor\delta' : \Delta'\rfloor\ \mathtt{g}(\delta') : \Gamma'$$

and a canonical homotopy equivalence $\langle\ f \mid g\ \rangle$:

$$\lfloor\gamma' : \Gamma', x : A(\gamma')\rfloor\ f(\gamma', x) : A'(\mathtt{f}(\gamma'))$$
$$\lfloor\gamma' : \Gamma', \underline{x}' : A'(\mathtt{f}(\gamma'))\rfloor\ g(\gamma', \underline{x}') : A(\gamma')$$

between $A(\gamma')$ and $A'(\delta')$ relative to $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$, in such a way that the extension:

$$\lfloor\gamma' : \Gamma', x : A(\gamma')\rfloor\ (\mathtt{f}, \mathtt{f}_{n+1})(\gamma', x) : \Delta', A'$$
$$\lfloor\delta' : \Delta', \underline{x} : A'(\delta')\rfloor\ (\mathtt{g}, \mathtt{g}_{n+1})(\delta', \underline{x}) : \Gamma', A$$

of $\langle\ \mathtt{f} \mid \mathtt{g}\ \rangle$ via $\langle\ f \mid g\ \rangle$ (Lemma 3.2) is the given $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$. Therefore $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ cannot be of the form `(1)` as (e.g.) $\delta : \Delta$ is not the empty context, which means that $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$, being canonical, is of the form `(2)`. Hence, as before, there are a canonical context homotopy equivalence $\langle\ \mathtt{f}' \mid \mathtt{g}'\ \rangle$:

$$\lfloor\gamma' : \Gamma'\rfloor\ \mathtt{f}'(\gamma') : \Delta'$$
$$\lfloor\delta' : \Delta'\rfloor\ \mathtt{g}'(\delta') : \Gamma'$$

and a canonical homotopy equivalence $\langle\ f' \mid g'\ \rangle$:

$$\lfloor\gamma' : \Gamma', x : A(\gamma')\rfloor\ f'(\gamma', x) : A'(\mathtt{f}'(\gamma'))$$
$$\lfloor\gamma' : \Gamma', \underline{\underline{x}}' : A'(\mathtt{f}'(\gamma'))\rfloor\ g'(\gamma', \underline{\underline{x}}') : A(\gamma')$$

between $A(\gamma')$ and $A'(\delta')$ relative to $\langle\ \mathtt{f}'\mid \mathtt{g}'\ \rangle$, in such a way that the extension:

$$\lfloor \gamma':\Gamma', x:A(\gamma')\rfloor\ (\mathtt{f}',\mathtt{f}'_{n+1})(\gamma',x):\Delta',A'$$
$$\lfloor \delta':\Delta', \underline{x}:A'(\delta')\rfloor\ (\mathtt{g}',\mathtt{g}'_{n+1})(\delta',\underline{x}):\Gamma',A$$

of $\langle\ \mathtt{f}'\mid \mathtt{g}'\ \rangle$ via $\langle\ f'\mid g'\ \rangle$ (Lemma 3.2) is the given $\langle\ \mathtt{c}'\mid \mathtt{d}'\ \rangle$. Now $\lfloor \gamma':\Gamma'\rfloor\ \alpha(\gamma'):\mathtt{f}(\gamma')=\mathtt{f}'(\gamma')$ (i.e. $\lfloor \delta':\Delta'\rfloor\ \tilde{\alpha}(\delta'):\mathtt{g}(\delta')=\mathtt{g}'(\delta')$) by inductive hypothesis.

Let us pretend that we know that:

$$\lfloor \gamma',x\rfloor\ f'(\gamma',x)=\alpha(\gamma')^* f'(\gamma',x)$$

(i.e. $\lfloor \gamma',\underline{\underline{x}}'\rfloor\ g(\gamma',\alpha(\gamma')^*\underline{\underline{x}}')=g'(\gamma',\underline{\underline{x}}')$), that is:

$$\lfloor \gamma',x\rfloor\ \mathtt{f}_{n+1}(\gamma',x)=\alpha(\gamma')^*\mathtt{f}'_{n+1}(\gamma',x)\ \text{(i.e. } \lfloor \delta',\underline{x}\rfloor\ \mathtt{g}_{n+1}(\delta',\underline{x})=\tilde{\alpha}(\delta')^*\mathtt{g}'_{n+1}(\delta',\underline{x})\text{)}.$$

Then:

$$\lfloor \gamma',x\rfloor\ (\mathtt{f},\mathtt{f}_{n+1})(\gamma',x)=(\mathtt{f}',\mathtt{f}'_{n+1})(\gamma',x)$$

(i.e. $\lfloor \delta',\underline{x}\rfloor\ (\mathtt{g},\mathtt{g}_{n+1})(\delta',\underline{x})=(\mathtt{g}',\mathtt{g}'_{n+1})(\delta',\underline{x})$) that is:

$$\lfloor \gamma:\Gamma\rfloor\ \mathtt{c}(\gamma)=\mathtt{c}'(\gamma)$$

(i.e. $\lfloor \delta:\Delta\rfloor\ \mathtt{d}(\delta)=\mathtt{d}'(\delta)$).

Hence we are done if $\lfloor \gamma',x\rfloor\ f'(\gamma',x)=\alpha(\gamma')^* f'(\gamma',x)$ (i.e. $\lfloor \gamma',\underline{\underline{x}}'\rfloor\ g(\gamma',\alpha(\gamma')^*\underline{\underline{x}}')=g'(\gamma',\underline{\underline{x}}')$), but this is actually the case by the following Lemma 4.18. □

Lemma 4.18. *Let $\gamma:\Gamma$ and $\delta:\Delta$ be contexts together with context homotopy equivalences $(\mathtt{f};\mathtt{g})$ and $(\mathtt{f}';\mathtt{g}')$ such that:*

$$\lfloor \gamma:\Gamma\rfloor\ \alpha(\gamma):\mathtt{f}(\gamma)=\mathtt{f}'(\gamma)\ \textit{(i.e. } \lfloor \delta:\Delta\rfloor\ \tilde{\alpha}(\delta):\mathtt{g}(\delta)=\mathtt{g}'(\delta)\textit{)}$$

and let $\lfloor \gamma:\Gamma\rfloor\ S(\gamma):\textsc{Type}$ and $\lfloor \delta:\Delta\rfloor\ T(\delta):\textsc{Type}$ have h-propositional identities. Moreover, let $\langle\ \varphi\mid\psi\ \rangle$ be a canonical homotopy equivalence between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f};\mathtt{g})$ and let $\langle\ \varphi'\mid\psi'\ \rangle$ be a canonical homotopy equivalence between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}';\mathtt{g}')$. Then $\lfloor \gamma,s:S(\gamma)\rfloor\ \varphi(\gamma,s)=\alpha(\gamma)^*\varphi'(\gamma,s)$ i.e. $\lfloor \gamma,\underline{t}':T(\mathtt{f}'(\gamma))\rfloor\ \psi(\gamma,\alpha(\gamma)^*\underline{t}')=\psi'(\gamma,\underline{t}')$.

Proof. By induction on the complexity of $S(\gamma)$.

If $S(\gamma)$ is a base type S then, being $\langle\ \varphi\mid\psi\ \rangle$ a canonical homotopy equivalence, the context δ needs to be γ and $T(\delta)$ needs to be S, hence both $\langle\ \varphi\mid\psi\ \rangle$ and $\langle\ \varphi'\mid\psi'\ \rangle$ need to be of the form `(a)` and we are done.

If $S(\gamma)$ is of the form $\Pi_{x:A(\gamma)}B(\gamma,x)$ for some $\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}$ with h-propositional identities and some $\lfloor\gamma,x\rfloor\ B(\gamma,x):\textsc{Type}$ with h-propositional identities then, being $\langle\ \varphi\mid\psi\ \rangle$ a canonical homotopy equivalence, the type $T(\delta)$ needs to be of the form $\Pi_{x':A'(\delta)}B'(\delta,x')$ for some $\lfloor\delta\rfloor\ A'(\delta):\textsc{Type}$ with h-propositional identities and some $\lfloor\delta,x'\rfloor\ B(\delta,x'):\textsc{Type}$ with h-propositional identities. Moreover, there are a canonical homotopy equivalence $\langle\ f_1\mid g_1\ \rangle$:

$$\lfloor \gamma:\Gamma, x:A(\gamma)\rfloor\ f_1(\gamma,x):A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma:\Gamma, \underline{x}':A'(\mathtt{f}(\gamma))\rfloor\ g_1(\gamma,\underline{x}'):A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f};\mathtt{g})$ and a canonical homotopy equivalence $\langle\ f_2\mid g_2\ \rangle$:

$$\lfloor \gamma:\Gamma, x:A(\gamma), y:B(\gamma,x)\rfloor\ f_2(\gamma,x,y):B'(\mathtt{f}(\gamma),f_1(\gamma,x))$$
$$\lfloor \gamma:\Gamma, x:A(\gamma), \underline{y}':B'(\mathtt{f}(\gamma),f_1(\gamma,x))\rfloor\ g_2(\gamma,x,\underline{y}'):B(\gamma,x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}, \mathtt{f}_{m+1}; \mathtt{g}, \mathtt{g}_{n+1})$ of $(\mathtt{f}; \mathtt{g})$ via $(f_1; g_1)$ (see Lemma 3.2), such that the homotopy equivalence $(f^{\Pi}; g^{\Pi})$ of Lemma 3.3 is the given $\langle\, \varphi \mid \psi \,\rangle$. Analogously, there are a canonical homotopy equivalence $\langle\, f'_1 \mid g'_1 \,\rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f'_1(\gamma, x) : A'(\mathtt{f}'(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}'(\gamma)) \rfloor\ g'_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}'; \mathtt{g}')$ and a canonical homotopy equivalence $\langle\, f'_2 \mid g'_2 \,\rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f'_2(\gamma, x, y) : B'(\mathtt{f}'(\gamma), f'_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}'(\gamma), f'_1(\gamma, x)) \rfloor\ g'_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma, x)$ and $B'(\delta, \underline{x})$ relative to the extension $(\mathtt{f}', \mathtt{f}'_{m+1}; \mathtt{g}', \mathtt{g}'_{n+1})$ of $(\mathtt{f}'; \mathtt{g}')$ via $(f'_1; g'_1)$ (see Lemma 3.2), such that the homotopy equivalence $(f'^{\Pi}; g'^{\Pi})$ of Lemma 3.3 is the given $\langle\, \varphi' \mid \psi' \,\rangle$. By inductive hypothesis:

$$\lfloor \gamma, x : A(\gamma) \rfloor\ \alpha_1(\gamma, x) : f_1(\gamma, x) = \alpha(\gamma)^* f'_1(\gamma, x) \text{ i.e.}$$
$$\lfloor \gamma, \underline{x}' : A'(\mathtt{f}'(\gamma)) \rfloor\ g_1(\gamma, \alpha(\gamma)^* \underline{x}') = g'_1(\gamma, \underline{x}')$$

and:

$$\lfloor \gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) = (\alpha(\gamma), \alpha_1(\gamma, x))^* f'_2(\gamma, x, y)$$
$$\text{i.e. } \lfloor \gamma, x : A(\gamma), \underline{y}' : B(\mathtt{f}'(\gamma), f'_1(\gamma, x)) \rfloor\ g_2(\gamma, x, (\alpha(\gamma), \alpha_1(\gamma, x))^* \underline{y}') = g'_2(\gamma, x, \underline{y}').$$

Then:

$$\begin{aligned} g_2(\gamma, x, \mathsf{ev}(\alpha(\gamma)^* \underline{z}', f_1(\gamma, x))) &= g_2(\gamma, x, \alpha_1(\gamma, x)^* \mathsf{ev}(\alpha(\gamma)^* \underline{z}', \alpha(\gamma)^* f'_1(\gamma, x))) \\ &= g_2(\gamma, x, \alpha_1(\gamma, x)^* (\alpha(\gamma), \mathsf{r}(\alpha(\gamma)^* f'_1(\gamma, x)))^* \mathsf{ev}(\underline{z}', f'_1(\gamma, x))) \\ &= g_2(\gamma, x, \alpha_1(\gamma, x)^* \alpha(\gamma)^* \mathsf{ev}(\underline{z}', f'_1(\gamma, x))) \\ &= g_2(\gamma, x, (\alpha(\gamma), \alpha_1(\gamma, x))^* \mathsf{ev}(\underline{z}', f'_1(\gamma, x))) \\ &= g'_2(\gamma, x, \mathsf{ev}(\underline{z}', f'_1(\gamma, x))) \end{aligned}$$

in context $\gamma, \underline{z}' : [\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}'(\gamma))$, where the first and the fourth equalities follows by (generalised) path induction on $\alpha_1(\gamma, x)$ and the second and the third by multiple (generalised) path induction on $\alpha(\gamma)$. By propositional function extensionality, we deduce that:

$$\lfloor \gamma, \underline{z}' : [\Pi_{\underline{x}:A'(\delta)} B'(\delta, \underline{x})](\mathtt{f}'(\gamma)) \rfloor\ g^{\Pi}(\gamma, \alpha(\gamma)^* \underline{z}') = g'^{\Pi}(\gamma, \underline{z}')$$

and we are done.

If $S(\gamma)$ is of the form $\Sigma_{x:A(\gamma)} B(\gamma, x)$ for some $\lfloor \gamma \rfloor\ A(\gamma) : \textsc{Type}$ with h-propositional identities and some $\lfloor \gamma, x \rfloor\ B(\gamma, x) : \textsc{Type}$ with h-propositional identities then, being $\langle\, \varphi \mid \psi \,\rangle$ a canonical homotopy equivalence, the type $T(\delta)$ needs to be of the form $\Sigma_{x':A'(\delta)} B'(\delta, x')$ for some $\lfloor \delta \rfloor\ A'(\delta) : \textsc{Type}$ with h-propositional identities and some $\lfloor \delta, x' \rfloor\ B(\delta, x') : \textsc{Type}$ with h-propositional identities. Moreover, there are a canonical homotopy equivalence $\langle\, f_1 \mid g_1 \,\rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma) \rfloor\ f_1(\gamma, x) : A'(\mathtt{f}(\gamma))$$
$$\lfloor \gamma : \Gamma, \underline{x}' : A'(\mathtt{f}(\gamma)) \rfloor\ g_1(\gamma, \underline{x}') : A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$ and a canonical homotopy equivalence $\langle\, f_2 \mid g_2 \,\rangle$:

$$\lfloor \gamma : \Gamma, x : A(\gamma), y : B(\gamma, x) \rfloor\ f_2(\gamma, x, y) : B'(\mathtt{f}(\gamma), f_1(\gamma, x))$$
$$\lfloor \gamma : \Gamma, x : A(\gamma), \underline{y}' : B'(\mathtt{f}(\gamma), f_1(\gamma, x)) \rfloor\ g_2(\gamma, x, \underline{y}') : B(\gamma, x)$$

between $B(\gamma,x)$ and $B'(\delta,\underline{x})$ relative to the extension $(\mathtt{f},\mathtt{f}_{m+1};\mathtt{g},\mathtt{g}_{n+1})$ of $(\mathtt{f};\mathtt{g})$ via $(f_1;g_1)$ (Lemma 3.2), such that the homotopy equivalence $(f^\Sigma;g^\Sigma)$ of Lemma 3.4 is the given $\langle\, \varphi \mid \psi \,\rangle$. Analogously, there are a canonical homotopy equivalence $\langle\, f_1' \mid g_1' \,\rangle$:

$$\lfloor \gamma:\Gamma, x:A(\gamma)\rfloor\ f_1'(\gamma,x):A'(\mathtt{f}'(\gamma))$$
$$\lfloor \gamma:\Gamma, \underline{x}':A'(\mathtt{f}'(\gamma))\rfloor\ g_1'(\gamma,\underline{x}'):A(\gamma)$$

between $A(\gamma)$ and $A'(\delta)$ relative to $(\mathtt{f}';\mathtt{g}')$ and a canonical homotopy equivalence $\langle\, f_2' \mid g_2' \,\rangle$:

$$\lfloor \gamma:\Gamma, x:A(\gamma), y:B(\gamma,x)\rfloor\ f_2'(\gamma,x,y):B'(\mathtt{f}'(\gamma),f_1'(\gamma,x))$$
$$\lfloor \gamma:\Gamma, x:A(\gamma), \underline{y}':B'(\mathtt{f}'(\gamma),f_1'(\gamma,x))\rfloor\ g_2'(\gamma,x,\underline{y}'):B(\gamma,x)$$

between $B(\gamma,x)$ and $B'(\delta,\underline{x})$ relative to the extension $(\mathtt{f}',\mathtt{f}_{m+1}';\mathtt{g}',\mathtt{g}_{n+1}')$ of $(\mathtt{f}';\mathtt{g}')$ via $(f_1';g_1')$ (Lemma 3.2), such that the homotopy equivalence $(f'^\Sigma;g'^\Sigma)$ of Lemma 3.4 is the given $\langle\, \varphi' \mid \psi' \,\rangle$. By inductive hypothesis:

$$\lfloor \gamma, x:A(\gamma)\rfloor\ \alpha_1(\gamma,x):f_1(\gamma,x)=\alpha(\gamma)^* f_1'(\gamma,x)$$

and:

$$\lfloor \gamma, x:A(\gamma), y:B(\gamma,x)\rfloor\ f_2(\gamma,x,y)=(\alpha(\gamma),\alpha_1(\gamma,x))^* f_2'(\gamma,x,y)$$

hence:

$$\begin{aligned}\lfloor \gamma, u:\Sigma_{x:A(\gamma)}B(\gamma,x)\rfloor\ f^\Sigma(\gamma,u) &\equiv \langle f_1(\gamma,\pi_1 u), f_2(\gamma,\pi_1 u,\pi_2 u)\rangle\\ &= \langle \alpha(\gamma)^* f_1'(\gamma,\pi_1 u), \alpha(\gamma)^* f_2'(\gamma,\pi_1 u,\pi_2 u)\rangle\\ &= \alpha(\gamma)^*\langle f_1'(\gamma,\pi_1 u), f_2'(\gamma,\pi_1 u,\pi_2 u)\rangle\\ &\equiv \alpha(\gamma)^* f'^\Sigma(\gamma,u)\end{aligned}$$

where the second propositional equality follows by multiple (generalised) path induction on $\alpha(\gamma)$. We are done.

If $S(\gamma)$ is an h-proposition $s_1(\gamma)=s_2(\gamma)$ for some $\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}$ with h-propositional identities and some $\lfloor\gamma\rfloor\ s_1(\gamma),s_2(\gamma):A(\gamma)$ then, being $\langle\, \varphi \mid \psi \,\rangle$ a canonical homotopy equivalence, the type $T(\delta)$ needs to be an h-proposition $t_1(\delta)=t_2(\delta)$ for some $\lfloor\delta\rfloor\ A'(\delta):\textsc{Type}$ with h-propositional identities and some $\lfloor\delta\rfloor\ t_1(\delta),t_2(\delta):A'(\delta)$. We observe that $\varphi(\gamma,p):t_1(\mathtt{f}(\gamma))=t_2(\mathtt{f}(\gamma))$ and $\varphi'(\gamma,p):t_1(\mathtt{f}'(\gamma))=t_2(\mathtt{f}'(\gamma))$ in context $\gamma,p:s_1(\gamma)=s_2(\gamma)$, hence $\alpha(\gamma)^*\varphi'(\gamma,p):t_1(\mathtt{f}(\gamma))=t_2(\mathtt{f}(\gamma))$. Since $t_1(\mathtt{f}(\gamma))=t_2(\mathtt{f}(\gamma))$ an h-proposition, then $\lfloor\gamma,p\rfloor\ \varphi(\gamma,p)=\alpha(\gamma)^*\varphi'(\gamma,p)$ and we are done. □

Remark 4.19. We observe that the soundness of the last paragraph of the previous proof is the key reason why we need to work in the restricted family of type judgements of Definition 4.13.

In Section 5 a further restriction of the contexts of PTT is adopted.

5. Making the syntax of Propositional Type Theory into a category with attributes

In this section we use the syntax of PTT in order to define a model of ETT. We remind that a type judgement $\lfloor\delta\rfloor\ T(\delta) : \text{Type}$ is said to be an *h-set* if every type judgement $\lfloor\delta, x_1, x_2 : T(\delta)\rfloor\ x_1 = x_2 : \text{Type}$ is an h-proposition. We start by defining a further restriction on the type family that we allow to build contexts:

Definition 5.1. A type judgement $\lfloor\delta\rfloor\ T(\delta) : \text{Type}$ in some context $\delta : \Delta$ is **h-elementary** if it belongs to the smallest family $\mathcal{F}$ of type judgements that satisfies the following clauses:

- a judgement $\lfloor\gamma\rfloor\ S : \text{Type}$ (where S is an atomic type) belongs to $\mathcal{F}$, whenever S is an h-set;
- a judgement $\lfloor\gamma\rfloor\ \Pi_{x:A(\gamma)}B(\gamma, x) : \text{Type}$, for some $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ of $\mathcal{F}$ and some $\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}$ of $\mathcal{F}$, belongs to $\mathcal{F}$;
- a judgement $\lfloor\gamma\rfloor\ \Sigma_{x:A(\gamma)}B(\gamma, x) : \text{Type}$, for some $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ of $\mathcal{F}$ and some $\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}$ of $\mathcal{F}$, belongs to $\mathcal{F}$;
- a judgement $\lfloor\gamma\rfloor\ s_1(\gamma) = s_2(\gamma) : \text{Type}$, for some $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ of $\mathcal{F}$ and some $\lfloor\gamma\rfloor\ s_1(\gamma), s_2(\gamma) : A(\gamma)$, belongs to $\mathcal{F}$.

The h-elementary types and the types with h-propositional identities (see Definition 4.13) differ in their restrictions: while the latter only restrict the identity type formation to when it produces h-propositions, h-elementary types further restrict the atomic types to the h-sets.

Definition 5.2. Let $\gamma : \Gamma$ be a context $\gamma_1 : \Gamma_1, \gamma_2 : \Gamma_2(\gamma_1), ..., \gamma_n : \Gamma_n(\gamma_1, ..., \gamma_{n-1})$, where n might be 0. For any $i \in \{1, ..., n\}$, let γ^i be the context $\gamma_1, ..., \gamma_i$. We say that γ is **h-elementary** if, for every $i \in \{1, ..., n\}$, the judgement $\lfloor\gamma^i\rfloor\ \Gamma_i(\gamma^i)$ is h-elementary.

We recall that in an intensional type theory the family of the h-sets is closed under Π, Σ, and $=$ operations. The same holds in PTT, i.e. whenever $\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}$ and $\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}$ are h-sets, the judgements:

$$\lfloor\gamma\rfloor\ \Sigma_{x:A(\gamma)}B(x, \gamma),\ \lfloor\gamma\rfloor\ \Pi_{x:A(\gamma)}B(x, \gamma), \text{ and } \lfloor\gamma, x, y : A(\gamma)\rfloor\ x = y$$

are h-sets as well. The proof works formally as in the intensional case. We refer the reader to [RS15, Section 2.3] and to [Uni13, Chapter 3]. Therefore, by induction on the complexity of an h-elementary type, we infer that:

Corollary 5.3. *Every h-elementary type is an h-set.*

and therefore, again by induction on the complexity of an h-elementary type, one proves that:

Remark 5.4. Every h-elementary type has h-propositional identities, hence every h-elementary context is a context with h-propositional identities. In particular, the results of Subsection 4.2 continue being true for the family of h-elementary types and contexts.

By induction on the complexity of a canonical homotopy equivalence, one proves that:

Lemma 5.5. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be h-elementary contexts together with a context homotopy equivalence $(\mathtt{f}; \mathtt{g})$ such that $|\gamma : \Gamma| \equiv |\delta : \Delta|$ and:*

$$\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |\mathtt{f}(\gamma)| \equiv |\gamma| \quad (\text{ i.e. } \lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |\mathtt{g}(\delta)| \equiv |\gamma|\ )$$

and let $\lfloor\gamma : \Gamma\rfloor\ S(\gamma) : \text{Type}$ and $\lfloor\delta : \Delta\rfloor\ T(\delta) : \text{Type}$ be h-elementary. Moreover, let $\langle\ \varphi \mid \psi\ \rangle$ be a canonical homotopy equivalence between $S(\gamma)$ and $T(\delta)$ relative to $(\mathtt{f}; \mathtt{g})$. Then $\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |S(\gamma)| \equiv |T(\delta)| \quad (\equiv |T(\mathtt{f}(\gamma))|\ )$ and:

$$\lfloor |\gamma|, |s| \rfloor_{\mathtt{ext}}\ |\varphi(\gamma, s)| \equiv |s| \quad \textit{(i.e. } \lfloor |\gamma|, |t'| \equiv |t| \equiv |s| \rfloor_{\mathtt{ext}}\ |\psi(\delta, t')| \equiv |s| \textit{).}$$

and therefore, by induction on the complexity of a canonical context homotopy equivalence, one infers that:

Proposition 5.6. *Let $\gamma : \Gamma$ and $\delta : \Delta$ be h-elementary contexts together with a canonical context homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ from γ to δ. Then $|\gamma| \equiv |\delta|$ and $\lfloor |\gamma| \rfloor_{\mathtt{ext}}\ |\mathtt{c}(\gamma)| = |\gamma|$ i.e. $\lfloor |\gamma| \rfloor_{\mathtt{ext}}\ |\mathtt{d}(\delta)| = |\gamma|$.*

5.1. **The semantic context category.** We say that two h-elementary contexts of PTT are equivalent if there is a canonical context homotopy equivalence between them. This request defines an equivalence relation: its reflexivity follows by Proposition 4.2, its symmetry follows by Proposition 4.4 and its transitivity from Proposition 4.8.

From now on, whenever we speak about contexts and types of PTT, we will actually refer to *h-elementary* contexts and types of PTT.

Let us assume that we are given two (h-elementary) context morphisms $\gamma \xrightarrow{f(\gamma)} \delta$ and $\gamma' \xrightarrow{f'(\gamma')} \delta'$ (of PTT). We say that $f(\gamma)$ and $f'(\gamma')$ are equivalent if and only if:

(1) the contexts γ and γ' are equivalent;
(2) the contexts δ and δ' are equivalent;
(3) if $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is a canonical context homotopy equivalence $\gamma \to \gamma'$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ is one $\delta \to \delta'$, then:

$$\begin{array}{ccc} \gamma & \xrightarrow{\ f(\gamma)\ } & \delta \\ {\scriptstyle \mathtt{c}(\delta)}\downarrow & /\!\!/ & \downarrow{\scriptstyle \mathtt{c}'(\delta)} \\ \gamma' & \xrightarrow[\ f'(\gamma')\ ]{} & \delta' \end{array}$$

i.e. $\lfloor \gamma \rfloor\ \mathtt{c}'(f(\gamma)) = f'(\mathtt{c}(\delta))$, where we observe that this condition does not depend on the choice of $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ by Proposition 4.17 (we remind that h-elementary contexts have h-propositional identities).

Then we observe that:

Proposition 5.7. *There is a category $\mathcal{C}$ such that:*

- *the objects of $\mathcal{C}$ are the equivalence classes $[\gamma : \Gamma]$ of h-elementary contexts;*
- *the arrows $[\gamma : \Gamma] \to [\delta : \Delta]$ of $\mathcal{C}$ are the equivalence classes $[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta']$ of morphisms of contexts $\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'$ where $[\gamma' : \Gamma'] = [\gamma]$ and $[\delta' : \Delta'] = [\delta]$;*
- *if we are given arrows:*

$$[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : [\gamma : \Gamma] \to [\delta : \Delta] \textit{ and } [\lfloor \delta'' : \Delta'' \rfloor\ g(\delta'') : \Omega'] : [\delta : \Delta] \to [\omega : \Omega]$$

then, whenever $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is a canonical homotopy equivalence $\delta' \to \delta''$, the composition arrow:

$$[\lfloor \delta'' : \Delta'' \rfloor\ g(\delta'') : \Omega'] \circ [\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : [\gamma : \Gamma] \to [\omega : \Omega]$$

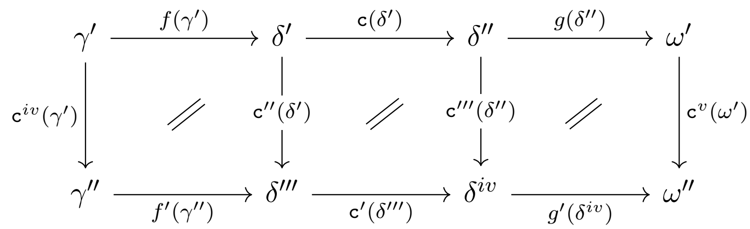
is the arrow $[\lfloor \gamma' : \Gamma' \rfloor\ g(\mathtt{c}(f(\gamma'))) : \Omega']$.

Moreover, the category $\mathcal{C}$ has a terminal object.

Proof. Observe that the composition operation is in fact well-defined: if we are given:

$$[\lfloor \gamma'' : \Gamma'' \rfloor\ f'(\gamma'') : \Delta'''] = [\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] \text{ and}$$
$$[\lfloor \delta^{iv} : \Delta^{iv} \rfloor\ g'(\delta^{iv}) : \Omega''] = [\lfloor \delta'' : \Delta'' \rfloor\ g(\delta'') : \Omega']$$

then:



for any choice of canonical context homotopy equivalences:

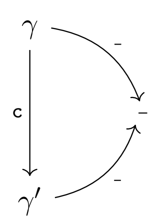
$$\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle : \delta''' \to \delta^{iv}$$
$$\langle\ \mathtt{c}'' \mid \mathtt{d}''\ \rangle : \delta' \to \delta'''$$
$$\langle\ \mathtt{c}''' \mid \mathtt{d}'''\ \rangle : \delta'' \to \delta^{iv}$$
$$\langle\ \mathtt{c}^{iv} \mid \mathtt{d}^{iv}\ \rangle : \gamma' \to \gamma''$$
$$\langle\ \mathtt{c}^{v} \mid \mathtt{d}^{v}\ \rangle : \omega' \to \omega''$$

where the propositional commutativity of the inner square is a consequence of Proposition 4.8 and Proposition 4.17. Hence:

$$[\lfloor \gamma'' : \Gamma'' \rfloor\ g'(\mathtt{c}'(f'(\gamma''))) : \Omega''] = [\lfloor \gamma' : \Gamma' \rfloor\ g(\mathtt{c}(f(\gamma'))) : \Omega']$$

and we are done. This operation is associative and the identity of $[\gamma : \Gamma]$ is the class $[\lfloor \gamma \rfloor\ \gamma]$, hence $\mathcal{C}$ is actually a category.

We observe that a terminal object of $\mathcal{C}$ is the class $[\ \_ : \_\ ]$ represented by the empty context. In fact, whenever we are given an object $[\gamma : \Gamma]$ of $\mathcal{C}$, an arrow $[\gamma : \Gamma] \to [\ \_ : \_\ ]$ is the one represented by the judgement $\lfloor \gamma : \Gamma \rfloor\ \_ : \_$. Moreover, it is unique: if we are given an arrow $[\gamma : \Gamma] \to [\ \_ : \_\ ]$ represented by some judgement $\lfloor \gamma' : \Gamma' \rfloor\ \_ : \_$, then the diagram:



commutes (for every $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ canonical $\gamma \to \gamma'$) even strictly by the terminality of $\_$ in the category of contexts of PTT, hence $\lfloor \gamma' \rfloor\ \_$ represents $[\lfloor \gamma \rfloor\ \_]$. □

We conclude the current subsection with the following:

Remark 5.8.

(1) By Proposition 4.2, if we are given two parallel morphisms of contexts $\lfloor \gamma \rfloor\ f(\gamma), f'(\gamma) : \Delta$, then $[\lfloor \gamma \rfloor\ f(\gamma)] = [\lfloor \gamma \rfloor\ f'(\gamma)]$ precisely when $\lfloor \gamma \rfloor\ f(\gamma) = f'(\gamma)$.

(2) By Proposition 4.4 and by Proposition 4.8, a judgement $\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Gamma''$ represents $[\lfloor \gamma \rfloor\ \gamma]$ (where $[\gamma'] = [\gamma] = [\gamma'']$) if and only if $\lfloor \gamma' \rfloor\ f(\gamma') = \mathtt{c}(\gamma')$ for some $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between γ' and γ''. In particular, if $\gamma' \equiv \gamma''$ then $\lfloor \gamma' \rfloor\ f(\gamma')$ represents $[\lfloor \gamma \rfloor\ \gamma]$ if and only if $\lfloor \gamma' \rfloor\ f(\gamma') = \gamma'$, by Proposition 4.2. Therefore $\lfloor \gamma' \rfloor\ \gamma'$ represents $[\lfloor \gamma \rfloor\ \gamma]$ whenever $[\gamma'] = [\gamma]$.

(3) If we are given an arrow $[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : [\gamma : \Gamma] \to [\delta : \Delta]$ then, if $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ are canonical between γ and γ' and between δ' and δ respectively, we infer by 1. (or by Proposition 4.4 and by Proposition 4.17) that:

$$[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] = [\lfloor \gamma : \Gamma \rfloor\ \mathtt{c}'(f(\mathtt{c}(\gamma))) : \Delta].$$

Hence without loss of generality we can always assume that we are given a representative of $[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta']$ of the form $\lfloor \gamma : \Gamma \rfloor\ f'(\gamma) : \Delta$.

(4) If we are given an arrow $[\gamma : \Gamma] \to [\delta : \Delta]$ represented by $\lfloor \gamma : \Gamma \rfloor\ f(\gamma) : \Delta$ and an arrow $[\delta : \Delta] \to [\omega : \Omega]$ represented by $\lfloor \delta : \Delta \rfloor\ g(\delta) : \Omega$, then their composition is represented by $g(\mathtt{c}(f(\gamma)))$ for any canonical homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between δ and itself. Hence $g(f(\gamma))$ represents their composition as well, by Proposition 4.2, Proposition 4.17 and 1. of Remark 5.8.

5.2. **The presheaf of semantic types.** In this subsection we define a presheaf of semantic types $\textsc{tp}$ associated to the category $\mathcal{C}$ of Proposition 5.7. If $\gamma : \Gamma$ and $\gamma' : \Gamma'$ are equivalent h-elementary contexts and $\lfloor \gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma' \rfloor\ A'(\gamma') : \textsc{Type}$ are h-elementary then we say that the judgements $\lfloor \gamma \rfloor\ A(\gamma) : \textsc{Type}$ and $\lfloor \gamma' \rfloor\ A'(\gamma') : \textsc{Type}$ are equivalent if there is a canonical homotopy equivalence between $A(\gamma)$ and $A'(\gamma')$ relative to some canonical context homotopy equivalence between γ and γ'. Equivalently (see Lemma 3.2 and the notion of canonical context homotopy equivalence), if the contexts $\gamma, x : A(\gamma)$ and $\gamma', x' : A(\gamma')$ are equivalent.

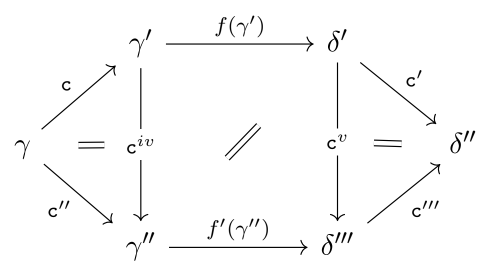
- If we are given an h-elementary context $\gamma : \Gamma$, we define $\textsc{tp}[\gamma : \Gamma]$ as the family of the classes $[\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}]$ where $[\gamma'] = [\gamma]$ and $\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}$ is h-elementary.
- If we are given an arrow $[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : [\gamma : \Gamma] \to [\delta : \Delta]$, we define the map $\textsc{tp}[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : \textsc{tp}[\delta : \Delta] \to \textsc{tp}[\gamma : \Gamma]$ as the one such that:

$$[\lfloor \delta'' \rfloor\ A(\delta'') : \textsc{Type}] \mapsto [\lfloor \gamma \rfloor\ A(\mathtt{c}'(f(\mathtt{c}(\gamma)))) : \textsc{Type}]$$

  where $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ are canonical between γ and γ' and between δ' and δ'' respectively. This relation is in fact a mapping:
  - Assuming that $[\lfloor \gamma'' : \Gamma'' \rfloor\ f'(\gamma'') : \Delta'''] = [\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta']$ and letting $\langle\ \mathtt{c}'' \mid \mathtt{d}''\ \rangle$ and $\langle\ \mathtt{c}''' \mid \mathtt{d}'''\ \rangle$ be canonical between γ and γ'' and between δ''' and δ'' respectively, we are left to verify that:

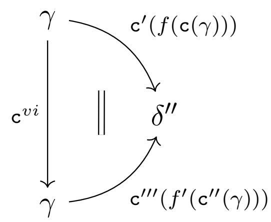
$$[\lfloor \gamma \rfloor\ A(\mathtt{c}'''(f'(\mathtt{c}''(\gamma)))) : \textsc{Type}] = [\lfloor \gamma \rfloor\ A(\mathtt{c}'(f(\mathtt{c}(\gamma)))) : \textsc{Type}].$$

    Let $\langle\ \mathtt{c}^{iv} \mid \mathtt{d}^{iv}\ \rangle$ and $\langle\ \mathtt{c}^{v} \mid \mathtt{d}^{v}\ \rangle$ be canonical between γ' and γ'' and between δ' and δ'''. Then:



    by Proposition 4.8. By Proposition 4.2, there is a canonical context homotopy equivalence $\langle\ \mathtt{c}^{vi} \mid \mathtt{d}^{vi}\ \rangle$ between γ and itself such that $\langle\ \mathtt{c}^{vi} \mid \mathtt{d}^{vi}\ \rangle$ is pairwise homotopic to

$(\lfloor\gamma\rfloor\ \gamma; \lfloor\gamma\rfloor\ \gamma)$. Hence:



and therefore we are done by Lemma 4.9.

– Assuming that $[\lfloor\delta'''\rfloor\ A'(\delta''') : \textsc{Type}] = [\lfloor\delta''\rfloor\ A(\delta'') : \textsc{Type}]$, and letting $\langle\ \mathtt{c}'' \mid \mathtt{d}''\ \rangle$ be canonical between δ' and δ''', we are left to verify that:

$$[\lfloor\gamma\rfloor\ A'(\mathtt{c}''(f(\mathtt{c}(\gamma)))) : \textsc{Type}] = [\lfloor\gamma\rfloor\ A(\mathtt{c}'(f(\mathtt{c}(\gamma)))) : \textsc{Type}].$$

Let $\langle\ \mathtt{c}''' \mid \mathtt{d}'''\ \rangle$ be a canonical homotopy equivalence between δ'' and δ'''. Then:

$$\begin{aligned}[\lfloor\delta'\rfloor\ A(\mathtt{c}'(\delta)) : \textsc{Type}] &= [\lfloor\delta''\rfloor\ A(\delta'') : \textsc{Type}]\\ &= [\lfloor\delta''\rfloor\ A'(\mathtt{c}'''(\delta'')) : \textsc{Type}]\\ &= [\lfloor\delta'\rfloor\ A'(\mathtt{c}''(\delta'))]\end{aligned}$$

by Corollary 4.10, by Lemma 4.11 and by Lemma 4.9 respectively. Then we are done by Lemma 4.12 with $\delta \equiv \delta'$ and $a(\gamma) \equiv f(\mathtt{c}(\gamma))$.

In the remainder of the current subsection we verify that:

Proposition 5.9. *The mapping* TP *is a functor.*

Proof.

- Let us consider the diagram:

$$[\gamma] \xrightarrow{[\lfloor\gamma'\rfloor\ f(\gamma'):\Delta']} [\delta] \xrightarrow{[\lfloor\delta''\rfloor\ g(\delta''):\Omega']} [\omega]$$

and let $[\lfloor\omega''\rfloor\ A(\omega'') : \textsc{Type}]$ be in $\textsc{tp}[\omega]$. Let us consider canonical context homotopy equivalences:

$$\begin{aligned}\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle &: \gamma \to \gamma'\\ \langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle &: \delta' \to \delta\\ \langle\ \mathtt{c}'' \mid \mathtt{d}''\ \rangle &: \delta \to \delta''\\ \langle\ \mathtt{c}''' \mid \mathtt{d}'''\ \rangle &: \omega' \to \omega'''\\ \langle\ \mathtt{c}^{iv} \mid \mathtt{d}^{iv}\ \rangle &: \delta' \to \delta''.\end{aligned}$$

Then, to verify that TP preserves the composition, it is enough to verify that:

$$[\lfloor\gamma\rfloor\ A(\mathtt{c}'''(g(\mathtt{c}''(\mathtt{c}'(f(\mathtt{c}(\gamma)))))))] = [\lfloor\gamma\rfloor\ A(\mathtt{c}'''(g(\mathtt{c}^{iv}(f(\mathtt{c}(\gamma))))))]$$

which is in fact true by Proposition 4.8 and Lemma 4.9.

- Let $[\gamma]$ be a context and let $[\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}]$ be in $\textsc{tp}[\gamma]$. As $[\gamma'] = [\gamma]$, the morphism $\lfloor\gamma'\rfloor\ \gamma'$ represents the identity of $[\gamma]$ (see 2. of Remark 5.8). Let us consider canonical context homotopy equivalences:

$$\begin{aligned}\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle &: \gamma' \to \gamma'\\ \langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle &: \gamma' \to \gamma'.\end{aligned}$$

Then $[\lfloor \gamma' \rfloor\ A(\mathtt{c}'(\mathtt{c}(\gamma')))] = [\lfloor \gamma' \rfloor\ A(\gamma')]$ by Corollary 4.10, hence $\textsc{tp}$ preserves the identities and we are done. □

We end the current subsection with the following:

Remark 5.10.

(1) By Corollary 4.10, if we are given a semantic type $[\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}]$ in semantic context $[\gamma : \Gamma]$, then:

$$[\lfloor \gamma \rfloor\ A(\mathtt{c}(\gamma)) : \textsc{Type}] = [\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}]$$

for every $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ canonical $\gamma \to \gamma'$. Hence we can always assume that we are given a representative of a semantic type in semantic context $[\gamma : \Gamma]$ whose context coincides with γ itself.

(2) If we are given an arrow $[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : [\gamma : \Gamma] \to [\delta : \Delta]$, we observe that the image of $[\lfloor \delta'' \rfloor\ A(\delta'') : \textsc{Type}]$ via $\textsc{tp}[\lfloor \gamma' : \Gamma' \rfloor\ f(\gamma') : \Delta'] : \textsc{tp}[\delta : \Delta] \to \textsc{tp}[\gamma : \Gamma]$, which is $[\lfloor \gamma \rfloor\ A(\mathtt{c}'(f(\mathtt{c}(\gamma)))) : \textsc{Type}]$ by definition, also coincides with:

$$[\lfloor \gamma' \rfloor\ A(\mathtt{c}'(f(\gamma'))) : \textsc{Type}]$$

by Corollary 4.10, being $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ and $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ canonical between γ and γ' and between δ' and δ''. We use this particular presentation in Subsection 5.3.

5.3. **The semantic context extension.** The semantic context extension $-.-$ associated to $\mathcal{C}$ and $\textsc{tp}$ (see Subsection 5.1 and Subsection 5.2 respectively) is defined as follows:

- An object $([\gamma : \Gamma], [\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}])$ of $\int \textsc{tp}$ is sent to:

$$[\gamma : \Gamma].[\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}] := [\gamma', x : A(\gamma')].$$

This mapping is well-defined because $[\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}] = [\lfloor \gamma'' \rfloor\ A'(\gamma'') : \textsc{Type}]$ precisely when $[\gamma', x : A(\gamma')] = [\gamma'', \underline{x} : A'(\gamma'')]$, by the notion of canonical context homotopy equivalence.

- An arrow $([\delta : \Delta], [\lfloor \delta' \rfloor\ B(\delta') : \textsc{Type}]) \to ([\gamma : \Gamma], [\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}])$ in $\int \textsc{tp}$ is an arrow $[\lfloor \delta'' : \Delta'' \rfloor\ f(\delta'') : \Gamma'']$ such that:

$$\textsc{tp}[\lfloor \delta'' : \Delta'' \rfloor\ f(\delta'') : \Gamma''] : [\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}] \mapsto [\lfloor \delta' \rfloor\ B(\delta') : \textsc{Type}].$$

Hence, if $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is a canonical context homotopy equivalence $\gamma'' \to \gamma'$, then:

$$[\lfloor \delta' \rfloor\ B(\delta') : \textsc{Type}] = [\lfloor \delta'' \rfloor\ A(\mathtt{c}(f(\delta''))) : \textsc{Type}]$$

by 2. or Remark 5.10. Moreover:

$$[\lfloor \gamma' \rfloor\ A(\gamma') : \textsc{Type}] = [\lfloor \gamma'' \rfloor\ A(\mathtt{c}(\gamma'')) : \textsc{Type}]$$

by Corollary 4.10. We stipulate that $-.-$ maps such an arrow of $\int \textsc{tp}$ to the arrow:

$$[\delta'', \underline{x} : A(\mathtt{c}(f(\delta'')))] \xrightarrow{[\ \lfloor \delta'', \underline{x}\ \rfloor\ f(\delta''), \underline{x}\ ]} [\gamma'', x : A(\mathtt{c}(\gamma''))].$$

Let us prove that this relation is actually a map. Let us consider a representative:

$$[\lfloor \delta''' : \Delta''' \rfloor\ f'(\delta''') : \Gamma'''] = [\lfloor \delta'' : \Delta'' \rfloor\ f(\delta'') : \Gamma'']$$

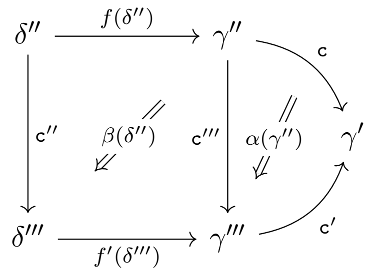
and let $\langle\ \mathtt{c}' \mid \mathtt{d}'\ \rangle$ be canonical $\gamma''' \to \gamma'$. We are left to verify that the corresponding:

$$[\delta''', \underline{x}' : A(\mathtt{c}'(f'(\delta''')))] \xrightarrow{[\ \lfloor \delta''', \underline{x}'\ \rfloor\ f'(\delta'''), \underline{x}'\ ]} [\gamma''', x' : A(\mathtt{c}'(\gamma'''))]$$

satisfies:

$$[\ \lfloor \delta'',\ \underline{x}\ \rfloor\ f(\delta''),\ \underline{x}\ ] = [\ \lfloor \delta''',\ \underline{x}'\ \rfloor\ f'(\delta'''),\ \underline{x}'\ ].$$

Let $\langle\ \mathsf{c}'' \mid \mathsf{d}''\ \rangle$ and $\langle\ \mathsf{c}''' \mid \mathsf{d}'''\ \rangle$ be canonical $\delta'' \to \delta'''$ and $\gamma'' \to \gamma'''$ and let us consider the diagram:



where $\lfloor \gamma'' \rfloor\ \alpha(\gamma'') : \mathsf{c}(\gamma'') = \mathsf{c}'(\mathsf{c}'''(\gamma''))$ exists by Proposition 4.8 and Proposition 4.17 and $\lfloor \delta'' \rfloor\ \beta(\delta'') : \mathsf{c}'''(f(\delta'')) = f'(\mathsf{c}''(\delta''))$ exists since $f(\delta'')$ and $f'(\delta''')$ represent the same arrow of $\mathcal{C}$. If:

$$\omega(\delta'') \equiv \alpha(f(\delta'')) \bullet \mathsf{c}'(\beta(\delta'')) : \mathsf{c}(f(\delta'')) = \mathsf{c}'(f'(\mathsf{c}''(\delta'')))$$

then by Lemma 4.9 there exists a canonical homotopy equivalence:

$$\langle\ \varphi \mid \psi\ \rangle$$

between $A(\mathsf{c}(f(\delta'')))$ and $A(\mathsf{c}'(f'(\delta''')))$ relative to $\langle\ \mathsf{c}'' \mid \mathsf{d}''\ \rangle$ and such that:

$$\lfloor \delta'', \underline{x} : A(\mathsf{c}(f(\delta'')))\rfloor\ \omega(\delta'')^{*}\varphi(\delta'', \underline{x}) = \underline{x}.$$

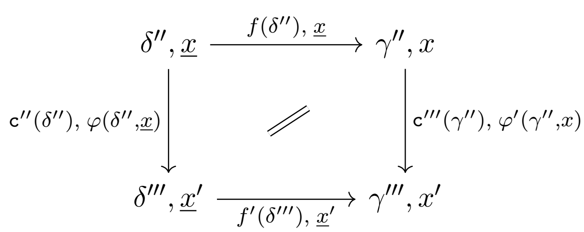
Anologously, again by Lemma 4.9 there is a canonical homotopy equivalence $\langle\ \varphi' \mid \psi'\ \rangle$ between $A(\mathsf{c}(\gamma''))$ and $A(\mathsf{c}'(\gamma'''))$ relative to $\langle\ \mathsf{c}'' \mid \mathsf{d}''\ \rangle$ and such that:

$$\lfloor \gamma'', x : A(\mathsf{c}(\gamma''))\rfloor\ \alpha(\gamma'')^{*}\varphi'(\gamma'', x) = x.$$

By the notion of canonical context homotopy equivalence and by Lemma 3.2, we obtain canonical context homotopy equivalences:

$$\lfloor \delta'', \underline{x} : A(\mathsf{c}(f(\delta'')))\rfloor\ \mathsf{c}''(\delta'') : \Delta''',\ \varphi(\delta'', \underline{x}) : A(\mathsf{c}'(f'(\mathsf{c}''(\delta''))))$$
$$\lfloor \gamma'', x : A(\mathsf{c}(\gamma''))\rfloor\ \mathsf{c}'''(\gamma'') : \Gamma''',\ \varphi'(\gamma'', x) : A(\mathsf{c}'(\mathsf{c}'''(\gamma'')))$$

between $\delta'', \underline{x} : A(\mathsf{c}(f(\delta'')))$ and $\delta''', \underline{x}' : A(\mathsf{c}'(f'(\delta''')))$ and between $\gamma'', x : A(\mathsf{c}(\gamma''))$ and $\gamma''', x' : A(\mathsf{c}'(\gamma'''))$ respectively. Therefore, we are left to verify that:



i.e. that:

$$\lfloor \delta'', \underline{x} \rfloor\ \mathsf{c}'''(f(\delta'')),\ \varphi'(f(\delta''), \underline{x}) = f'(\mathsf{c}''(\delta'')),\ \varphi(\delta'', \underline{x}).$$

Since $\lfloor \delta'' \rfloor\ \beta(\delta'') : \mathsf{c}'''(f(\delta'')) = f'(\mathsf{c}''(\delta''))$, we are left to verify that $\lfloor \delta'', \underline{x} \rfloor\ \varphi'(f(\delta''), \underline{x}) = \beta(\delta'')^{*}\varphi(\delta'', \underline{x})$. But since $\lfloor \delta'', \underline{x} \rfloor\ \beta(\delta'')^{*}\varphi(\delta'', \underline{x}) = \mathsf{c}'(\beta(\delta''))^{*}\varphi(\delta'', \underline{x})$ by multiple (generalised) path induction on $\beta(\delta'')$, we are left to verify that:

$$\lfloor \delta'', \underline{x} \rfloor\ \varphi'(f(\delta''), \underline{x}) = \mathsf{c}'(\beta(\delta''))^{*}\varphi(\delta'', \underline{x}).$$

This is equivalent to verifying that:

$$\lfloor \delta'', \underline{x} \rfloor\ \alpha(f(\delta''))^* \varphi'(f(\delta''), \underline{x}) = \alpha(f(\delta''))^* \mathsf{c}'(\beta(\delta''))^* \varphi(\delta'', \underline{x})$$

and this is true as $\alpha(f(\delta''))^* \varphi'(f(\delta''), \underline{x}) = \underline{x}$ and

$$\alpha(f(\delta''))^* \mathsf{c}'(\beta(\delta''))^* \varphi(\delta'', \underline{x}) = \omega(\delta'')^* \varphi(\delta'', \underline{x}) = \underline{x}.$$

We are left to verify that:

Proposition 5.11. *The mapping $-.-$ defines a functor.*

Proof.

- Suppose that we are given two arrows:

$$([\omega : \Omega], [\lfloor \omega' \rfloor\ C(\omega') : \text{Type}]) \xrightarrow{[g(\omega''):\Delta''']} ([\delta : \Delta], [\lfloor \delta' \rfloor\ B(\delta') : \text{Type}])$$
$$([\delta : \Delta], [\lfloor \delta' \rfloor\ B(\delta') : \text{Type}]) \xrightarrow{[f(\delta''):\Gamma'']} ([\gamma : \Gamma], [\lfloor \gamma' \rfloor\ A(\gamma') : \text{Type}])$$

  in $\int$ tp. We can rewrite the objects as follows:

$$([\gamma : \Gamma], [\lfloor \gamma' \rfloor\ A(\gamma') : \text{Type}]) = ([\gamma : \Gamma], [\lfloor \gamma'' \rfloor\ A(\mathsf{c}(\gamma'')) : \text{Type}])$$
$$([\delta : \Delta], [\lfloor \delta' \rfloor\ B(\delta') : \text{Type}]) = ([\delta : \Delta], [\lfloor \delta'' \rfloor\ A(\mathsf{c}(f(\delta''))) : \text{Type}])$$
$$([\omega : \Omega], [\lfloor \omega' \rfloor\ C(\omega') : \text{Type}]) = ([\omega : \Omega], [\lfloor \omega'' \rfloor\ A(\mathsf{c}(f(\mathsf{c}'(g(\omega''))))) : \text{Type}])$$

  for some $\langle\ \mathsf{c} \mid \mathsf{d}\ \rangle$ and $\langle\ \mathsf{c}' \mid \mathsf{d}'\ \rangle$ canonical $\gamma'' \to \gamma'$ and $\delta''' \to \delta''$ respectively. The composition of $\int$ tp sends this diagram to the arrow $[f(\mathsf{c}'(g(\omega''))) : \Gamma'']$ of the form:

$$([\omega : \Omega], [\lfloor \omega'' \rfloor\ A(\mathsf{c}(f(\mathsf{c}'(g(\omega''))))) : \text{Type}]) \to ([\gamma : \Gamma], [\lfloor \gamma'' \rfloor\ A(\mathsf{c}(\gamma'')) : \text{Type}]).$$

  Now, the arrows $[f(\delta'')]$, $[g(\omega'')]$ and $[f(\mathsf{c}'(g(\omega'')))]$ are sent by $-.-$ to the arrows:

$$[\delta'', \underline{x} : A(\mathsf{c}(f(\delta'')))] \xrightarrow{[f(\delta''), \underline{x}]} [\gamma'', x : A(\mathsf{c}(\gamma''))]$$
$$[\omega'', \underline{\underline{x}} : A(\mathsf{c}(f(\mathsf{c}'(g(\omega'')))))] \xrightarrow{[g(\omega''), \underline{\underline{x}}]} [\delta''', \underline{x}' : A(\mathsf{c}(f(\mathsf{c}'(\delta'''))))]$$
$$[\omega'', \underline{\underline{x}} : A(\mathsf{c}(f(\mathsf{c}'(g(\omega'')))))] \xrightarrow{[f(\mathsf{c}'(g(\omega''))), \underline{\underline{x}}]} [\gamma'', x : A(\mathsf{c}(\gamma''))]$$

  of $\mathcal{C}$ respectively. In order to conclude that $-.-$ actually preserves the composition, we are left to verify that the composition of the first and the second of these yields the third. By Corollary 4.10, there is $\langle\ \varphi \mid \psi\ \rangle$ canonical between $A(\mathsf{c}(f(\mathsf{c}'(\delta'''))))$ and $A(\mathsf{c}(f(\delta'')))$ relative to $\langle\ \mathsf{c}' \mid \mathsf{d}'\ \rangle$ and such that $\lfloor \delta''', \underline{x}' \rfloor\ \varphi(\delta''', \underline{x}') = \underline{x}'$. Hence, by Lemma 3.2 and by the notion of canonical context homotopy equivalence, there is a canonical context homotopy equivalence $\delta''', \underline{x}' \to \delta'', \underline{x}$ whose first component is $\mathsf{c}'(\delta'''), \varphi(\delta''', \underline{x}')$ and therefore the composition of $[f(\delta''), \underline{x}] \circ [g(\omega''), \underline{\underline{x}}]$ is represented by:

$$\lfloor \omega'', \underline{\underline{x}} \rfloor\ f(\mathsf{c}'(g(\omega''))), \varphi(g(\omega''), \underline{\underline{x}})$$

  which is in fact propositionally equal to $\lfloor \omega'', \underline{\underline{x}} \rfloor\ f(\mathsf{c}'(g(\omega''))), \underline{\underline{x}}$ since $\lfloor \omega'', \underline{\underline{x}} \rfloor\ \varphi(g(\omega''), \underline{\underline{x}}) = \underline{\underline{x}}$. We are done by 1. of Remark 5.8.
- By 2. of Remark 5.8, the identity over $([\gamma], [\lfloor \gamma' \rfloor\ A(\gamma') : \text{Type}])$ is represented by the judgement $\lfloor \gamma' \rfloor\ \gamma'$. By definition, it is sent by $-.-$ to the arrow:

$$[\gamma', x' : A(\mathsf{c}(\gamma'))] \xrightarrow{[\lfloor \gamma', x' \rfloor\ \gamma', x']} [\gamma', x' : A(\mathsf{c}(\gamma'))]$$

for some $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ canonical $\gamma' \to \gamma'$. This is in fact the identity of $[\gamma', x' : A(\mathtt{c}(\gamma'))] = [\gamma', x : A(\gamma')]$. Hence identities are preserved and we are done. □

5.4. **The display map family.** We define a cartesian natural transformation P from the semantic context extension and the projection, as functors $\int \mathrm{TP} \to \mathcal{C}$. Whenever $([\gamma], [\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}])$ is an object of $\int \mathrm{TP}$ then let the component of P in $([\gamma], [\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}])$ be the arrow:

$$[\gamma', x : A(\gamma')] \xrightarrow{[\lfloor\gamma', x\rfloor\ \gamma']} [\gamma'].$$

The mapping P happens to be well-defined: as long as $[\lfloor\gamma''\rfloor\ A'(\gamma'') : \textsc{Type}] = [\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}]$ then the diagram:

$$\begin{array}{ccc} \gamma', x & \xrightarrow{\quad\gamma'\quad} & \gamma' \\ {\scriptstyle \mathtt{c}(\gamma'), \varphi(\gamma', x)}\downarrow & & \downarrow{\scriptstyle \mathtt{c}(\gamma')} \\ \gamma'', \underline{x} & \xrightarrow[\quad\gamma''\quad]{} & \gamma'' \end{array}$$

commutes even strictly whenever $\langle\ \varphi \mid \psi\ \rangle$ is canonical $A(\gamma') \to A'(\gamma'')$ relative to some canonical $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between γ' and γ''. Let us verify that:

Proposition 5.12. *The family P is natural and cartesian.*

Proof.

- *Naturality.* Let us consider an arrow:

$$[\lfloor\delta'' : \Delta''\rfloor\ f(\delta'') : \Gamma''] : ([\delta : \Delta], [\lfloor\delta'\rfloor\ B(\delta') : \textsc{Type}]) \to ([\gamma : \Gamma], [\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}])$$

in $\int \mathrm{TP}$. By 3. of Remark 5.8, we can assume without loss of generality that $\delta'' \equiv \delta'$ and $\gamma'' \equiv \gamma'$. If $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ is canonical between γ' and itself, then:

$$[\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}] = [\lfloor\gamma'\rfloor\ A(\mathtt{c}(\gamma')) : \textsc{Type}]$$
$$[\lfloor\delta'\rfloor\ B(\delta') : \textsc{Type}] = [\lfloor\delta'\rfloor\ A(\mathtt{c}(f(\delta'))) : \textsc{Type}]$$

Then the images of $[\lfloor\delta'\rfloor\ f(\delta)' : \Gamma']$ via the semantic context extension and via the projection admit the presentations:

$$[\delta', x' : A(\mathtt{c}(f(\delta')))] \xrightarrow{[\lfloor\delta', x'\rfloor\ f(\delta'), x']} [\gamma', x : A(\mathtt{c}(\gamma'))]$$
$$[\delta'] \xrightarrow{[\lfloor\delta'\rfloor\ f(\delta')]} [\gamma']$$

and we are left to verify that they commute with P, in order to conclude the naturality of P itself. By considering the representatives $\lfloor\delta', x'\rfloor\ \delta'$ and $\lfloor\gamma', x\rfloor\ \gamma'$ of the $[\delta', x']$-component and the $[\gamma', x]$-component of P respectively, we are done by 1. and 4. of Remark 5.8.

- *Cartesianity.* Again, let us consider an arrow:

$$[\lfloor\delta' : \Delta'\rfloor\ f(\delta') : \Gamma'] : ([\delta : \Delta], [\lfloor\delta'\rfloor\ B(\delta') : \textsc{Type}]) \to ([\gamma : \Gamma], [\lfloor\gamma'\rfloor\ A(\gamma') : \textsc{Type}])$$

in $\int$ TP (the choice of its representative is justified by 3. of Remark 5.8) and let us verify that the commutative square:

$$\begin{array}{ccc} [\delta', x'] & \xrightarrow{[f(\delta'), x']} & [\gamma', x] \\ {\scriptstyle [\delta']}\downarrow & & \downarrow{\scriptstyle [\gamma']} \\ [\delta'] & \xrightarrow{[f(\delta')]} & [\gamma'] \end{array}$$

of $\mathcal{C}$ is a pullback of $\mathcal{C}$, being $x' : A(\mathsf{c}(f(\delta')))$ and $x : A(\mathsf{c}(\gamma'))$ and being $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ canonical $\gamma \to \gamma'$. Hence, let us assume that we are given two arrows $\chi : [\omega] \to [\delta']$ and $\chi' : [\omega] \to [\gamma', x]$ of $\mathcal{C}$ such that:

$$\begin{array}{ccc} [\omega] & \longrightarrow & [\gamma', x] \\ \downarrow & & \downarrow{\scriptstyle [\gamma']} \\ [\delta'] & \xrightarrow{[f(\delta')]} & [\gamma'] \end{array}$$

commutes. By 3. of Remark 5.8 there are representatives of the form:

$$\lfloor \omega \rfloor\ \alpha_1(\omega) : \Delta' \text{ and } \lfloor \omega \rfloor\ \alpha_2(\omega) : \Gamma', \alpha_3(\omega) : A(\mathsf{c}(\alpha_2(\omega)))$$

of χ and χ' respectively and $\lfloor \omega \rfloor\ p(\omega) : f(\alpha_1(\omega)) = \alpha_2(\omega)$, by 1. and 4. of Remark 5.8. Therefore $p(\omega)^*\alpha_3(\omega) : A(\mathsf{c}(f(\alpha_1(\omega))))$ and we obtain a morphism of contexts:

$$\lfloor \omega \rfloor\ \alpha_1(\omega) : \Delta',\ p(\omega)^*\alpha_3(\omega) : A(\mathsf{c}(f(\alpha_1(\omega))))$$

which represents a morphism of semantic contexts $\underline{\chi} : [\omega] \to [\delta', x']$ in $\mathcal{C}$. Post-composing $\underline{\chi}$ via $[\delta', x'] \xrightarrow{[\delta']} [\delta']$ and via $[\delta', x'] \xrightarrow{[f(\delta'), x']} [\gamma', x]$ we get morphisms $[\omega] \to [\delta']$ and $[\omega] \to [\gamma', x]$ respectively represented by:

$$\lfloor \omega \rfloor\ \alpha_1(\omega) : \Delta' \text{ and } \lfloor \omega \rfloor\ f(\alpha_1(\omega)) : \Gamma',\ p(\omega)^*\alpha_3(\omega) : A(\mathsf{c}(f(\alpha_1(\omega))))$$

respectively (by 4. of Remark 5.8). The former is clearly χ, while the latter is χ' because:

$$\lfloor \omega \rfloor\ p(\omega) : f(\alpha_1(\omega)) = \alpha_2(\omega)$$
$$\lfloor \omega \rfloor\ p(\omega)^*\alpha_3(\omega) \equiv p(\omega)^*\alpha_3(\omega)$$

which means (1. of Remark 5.8) that the morphisms:

$$f(\alpha_1(\omega)),\ p(\omega)^*\alpha_3(\omega) \text{ and } \alpha_2(\omega), \alpha_3(\omega)$$

represent the same arrow of $\mathcal{C}$.

Now, let $\underline{\chi}'$ be a morphism $[\omega] \to [\delta', x']$ such that $[\delta']\underline{\chi}' = \chi$ and $[f(\delta'), x']\underline{\chi}' = \chi'$. By 3. of Remark 5.8, the morphism $\underline{\chi}'$ is represented by a morphism of contexts of the form:

$$\lfloor \omega \rfloor\ \beta_1(\omega) : \Delta',\ \beta_2(\omega) : A(\mathsf{c}(f(\beta_1(\omega)))).$$

Moreover:

$$\lfloor \omega \rfloor\ p_1(\omega) : \alpha_1(\omega) = \beta_1(\omega)$$
$$\lfloor \omega \rfloor\ p_2(\omega) : \alpha_2(\omega) = f(\beta_1(\omega))$$
$$\lfloor \omega \rfloor\ p_3(\omega) : \alpha_3(\omega) = p_2(\omega)^*\beta_2(\omega)$$

by 1. and 4. or Remark 5.8, hence $p_1(\omega) : \alpha_1(\omega) = \beta_1(\omega)$ and:

$$\begin{aligned} p(\omega)^*\alpha_3(\omega) &= p(\omega)^* p_2(\omega)^* \beta_2(\omega) \\ &= (p(\omega) \bullet p_2(\omega))^* \beta_2(\omega) \\ &= f(p_1(\omega))^* \beta_2(\omega) \\ &= p_1(\omega)^* \beta_2(\omega) \end{aligned}$$

where the first and the second equalities are instances of propositional functorialities, the fourth holds by multiple (generalised) path induction on $p_1(\omega)$ and the third because $\gamma' : \Gamma'$ is an h-elementary context and by Corollary 5.3. Therefore, the judgements $\alpha_1(\omega),\ p(\omega)^*\alpha_3(\omega)$ and $\beta_1(\omega), \beta_2(\omega)$ represent the same arrow $[\omega] \to [\delta', x']$ i.e. $\underline{\chi} = \underline{\chi}'$. □

Remark 5.13. The last part of the proof of the cartesianity of the natural transformation P is the crucial point where we needed to work with h-elementary contexts. Any other result that we obtained so far is true for the (generally larger) family of contexts with h-propositional identities.

We summarise the content of the current section into the following:

Theorem 5.14. *The category $\mathcal{C}$ of Proposition 5.7, together with the presheaf* TP *of Proposition 5.9, the semantic context extension $-.-$ of Proposition 5.11 and the natural transformation P of Proposition 5.12, forms a category with attributes $\mathfrak{C} := (\mathcal{C}, \text{TP}, -.-, P)$.*

We end the current section with the following:

Remark 5.15. A terminal object preserving functor from the base category of $\mathfrak{C}$ to the base category of **eTT** mapping:

$$(\ [\delta : \Delta] \xrightarrow{[\lfloor\delta\rfloor\ f(\delta):\Gamma]} [\gamma : \Gamma]\ ) \mapsto (\ |\delta| \xrightarrow{|\lfloor\delta\rfloor\ f(\delta):\Gamma|} |\gamma|\ ) = (\ |\delta| \xrightarrow{\lfloor|\delta|\rfloor_{\mathtt{ext}}\ |f(\delta)|:|\Gamma|} |\gamma|\ )$$

is well-defined by Proposition 5.6 and extends to a morphism of semantic types $\mathfrak{C} \to \mathbf{eTT}$ if we stipulate that:

$$[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]\ \mapsto\ |\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}|\ =\ \lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |A(\gamma)| : \text{Type}$$

(see Remark 2.23 for more details). Observe in fact that this operation is well-defined by Lemma 5.5 and is natural since $\lfloor|\gamma|\rfloor_{\mathtt{ext}}\ |A(f(\delta))| \equiv |A|(|f(\delta)|)$ by Remark 2.23 (here we are implicitly using that:

$$(\text{TP}[\lfloor\delta\rfloor\ f(\delta) : \Gamma])[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}] = [\lfloor\delta\rfloor\ A(f(\delta)) : \text{Type}]$$

and this presentation of the substitution in $\mathfrak{C}$ is justified by Proposition 4.2, Corollary 4.10 and Lemma 4.12). One can verify that the semantic context extension and the display map family are preserved.

Section 6 is devoted to proving that $\mathfrak{C}$, together with the structure specified in Theorem 5.14, is a model of ETT.

6. A model of Extensional Type Theory

In this section, we show that $\mathfrak{C}$ verifies the requirements of Definition 2.17 where T is ETT. We start by observing that, if T is an atomic type of ETT and $t : T$ is an atomic term of ETT, then T is an atomic h-set of PTT, hence the class $[\ \lfloor\rfloor\ T : \textsc{Type}\ ]$ is a semantic type of $\mathfrak{C}$ in semantic context $[\ \_ : \_\ ]$ and $[\ \lfloor\rfloor\ t : T\ ]$ is a section of the corresponding display map, hence a semantic term of $[\ \lfloor\rfloor\ T : \textsc{Type}\ ]$ in $\mathfrak{C}$. Therefore the mappings:

$$T \mapsto [\ \lfloor\rfloor\ T : \textsc{Type}\ ]$$
$$t \mapsto [\ \lfloor\rfloor\ t : T\ ]$$

define a choice function as the one required in Definition 2.17.

After studying an opportune presentation of the semantic terms of $\mathfrak{C}$ in Subsection 6.1, we show that $\mathfrak{C}$ has semantic extensional identity types, semantic dependent product types, and semantic dependent sum types. We present the proof for the extensional identities in full form, while we only leave a sketch of the corresponding ones for dependent products and sums. Finally, we deduce the conservativity result in Subsection 6.4.

6.1. A presentation of the sections in the quotient syntax.

Let $[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]$ be some semantic type in some semantic context $[\gamma : \Gamma]$ (by 1. of Remark 5.10 we do not lose generality if we assume this presentation for $[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]$). Then, its component of P admits the presentation $[\lfloor\gamma, x : A(\gamma)\rfloor\ \gamma]$. Let us consider a section:

$$[\gamma] \to [\gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}] = [\gamma, x]$$

of $[\lfloor\gamma, x\rfloor\ \gamma]$, that, by 3. of Remark 5.8, admits the presentation $[\lfloor\gamma\rfloor\ \tilde{a}(\gamma) : \Gamma, a(\gamma) : A(\tilde{a}(\gamma))]$. Then $\lfloor\gamma\rfloor\ \alpha(\gamma) : \gamma = \tilde{a}(\gamma)$, by 1. of Remark 5.8 and being $[\lfloor\gamma\rfloor\ \tilde{a}(\gamma), a(\gamma)]$ a section of $[\lfloor\gamma, x\rfloor\ \gamma]$. Hence the morphism of contexts:

$$\lfloor\gamma\rfloor\ \gamma, \alpha(\gamma)^{*}a(\gamma) : A(\gamma)$$

is context propositionally equal to $\lfloor\gamma\rfloor\ \tilde{a}(\gamma), a(\gamma)$ and therefore it continues representing the given section $[\gamma] \to [\gamma, x]$.

We conclude that:

Remark 6.1. Without loss of generality, every section of a display map $[\lfloor\gamma, x\rfloor\ \gamma]$ is of the form:

$$[\lfloor\gamma\rfloor\ \gamma, a(\gamma)]$$

for some term $\lfloor\gamma\rfloor\ a(\gamma) : A(\gamma)$.

6.2. Semantic extensional identity types.

Let $[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]$ be a semantic type in semantic context $[\gamma : \Gamma]$ (see 1. or Remark 5.10). A presentation of $[\gamma : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]$ is $[\gamma, x : A(\gamma)]$, hence:

$$(\textsc{tp}[\lfloor\gamma, x\rfloor\ \gamma])([\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]) = [\lfloor\gamma, x\rfloor\ A(\mathtt{c}(\gamma)) : \textsc{Type}]$$

for some canonical homotopy equivalence $\langle\ \mathtt{c} \mid \mathtt{d}\ \rangle$ between γ and itself. In particular:

$$(\textsc{tp}[\lfloor\gamma, x\rfloor\ \gamma])([\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]) = [\lfloor\gamma, x\rfloor\ A(\gamma) : \textsc{Type}]$$

by Propostion Proposition 4.2 and by Corollary 4.10. Therefore we obtain the presentation:

$$[\gamma, x : A(\gamma), y : A(\gamma)]$$

of $([\gamma : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]).((\text{tp}[\lfloor\gamma, x\rfloor\ \gamma])([\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]))$. We define the semantic type $\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}$ in context $[\gamma, x, y]$ as the one represented by the type judgement:

$$\lfloor\gamma, x, y\rfloor\ x = y : \text{Type}$$

and the morphism of contexts $\mathsf{r}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}$ between $[\gamma, x]$ and $[\gamma, x, y].[\lfloor\gamma, x, y\rfloor\ x = y : \text{Type}] = [\gamma, x, y, p : x = y]$ as the one represented by the context morphism:

$$\lfloor\gamma, x\rfloor\ \gamma, x, x, \mathsf{r}(x).$$

We verified that *formation* and *introduction* of Definition 2.11 are satisfied. We are left to verify that:

Proposition 6.2. *The remaining conditions of Definition 2.11—i.e.* extensionality *and* compatibility with the substitution*—are satisfied by the above choice of:*

$$\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]} \text{ and } \mathsf{r}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}$$

hence $\mathfrak{C}$ is equipped with semantic extensional identity types.

Proof.

- *Extensionality*. Let us consider two semantic terms:

$$[\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)] \text{ and } [\lfloor\gamma\rfloor\ \gamma : \Gamma, b(\gamma) : A(\gamma)]$$

of the semantic type $[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]$ in the semantic context $[\gamma : \Gamma]$. They admit such a presentation because of 3. of Remark 5.8 and Remark 6.1. Now, if we consider $[\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)]$ as an arrow of $\int \text{tp}$ of source $([\gamma], [\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}])$ and target $([\gamma, x], [\lfloor\gamma, x\rfloor\ A(\gamma) : \text{Type}])$, its semantic context extension:

$$[\lfloor\gamma\rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)].[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]$$

is:

$$[\gamma, x] \xrightarrow{[\lfloor\gamma\rfloor\ \gamma, a(\gamma), x]} [\gamma, x, y]$$

hence the arrow:

$$[\gamma] \xrightarrow{[\lfloor\gamma\rfloor\ \gamma, b(\gamma):A(\gamma)]} [\gamma, x] \xrightarrow{[\lfloor\gamma\rfloor\ \gamma:\Gamma, a(\gamma):A(\gamma)].[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]} [\gamma, x, y]$$

of Lemma 2.6 happens to be represented by $\lfloor\gamma\rfloor\ \gamma, a(\gamma), b(\gamma)$, because of 4. of Remark 5.8.
By Proposition 4.2, by Corollary 4.10 and by Lemma 4.12, the semantic type:

$$\text{tp}[\lfloor\gamma\rfloor\ \gamma, a(\gamma), b(\gamma)](\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}) = \text{tp}[\lfloor\gamma\rfloor\ \gamma, a(\gamma), b(\gamma)]([\lfloor\gamma, x, y\rfloor\ x = y : \text{Type}])$$

in semantic context $[\gamma]$ admits the presentation:

$$\lfloor\gamma\rfloor\ a(\gamma) = b(\gamma) : \text{Type}.$$

With this presentation, if we are given a semantic term of this semantic type in context $[\gamma]$, i.e. a section $[\gamma] \to [\gamma, p' : a(\gamma) = b(\gamma)]$ of the corresponding display map $[\lfloor\gamma, p' : a(\gamma) = b(\gamma)\rfloor\ \gamma]$, then by Remark 6.1 it admits a representative of the form $\lfloor\gamma\rfloor\ \gamma : \Gamma, p(\gamma) : a(\gamma) = b(\gamma)$ for some judgement $\lfloor\gamma\rfloor\ p(\gamma) : a(\gamma) = b(\gamma)$. Hence $[\lfloor\gamma\rfloor\ \gamma, a(\gamma)] = [\lfloor\gamma\rfloor\ \gamma, b(\gamma)]$, by 1. of Remark 5.8. We are left to verify that:

$$[\lfloor\gamma\rfloor\ \gamma : \Gamma, p(\gamma) : a(\gamma) = b(\gamma)] = \mathsf{r}^{[\lfloor\gamma\rfloor\ \gamma:\Gamma, a(\gamma):A(\gamma)]}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}.$$

As the diagram (where $p'' : a(\gamma) = a(\gamma)$):

$$\begin{array}{ccc} \gamma & \xrightarrow{\lfloor \gamma \rfloor\ \gamma, a(\gamma)} & \gamma, x \\ {\scriptstyle \lfloor \gamma \rfloor\ \gamma, \mathsf{r}(a(\gamma))} \downarrow & & \downarrow {\scriptstyle \lfloor \gamma, x \rfloor\ \gamma, x, x, \mathsf{r}(x)} \\ \gamma, p'' & \xrightarrow{\lfloor \gamma, p'' \rfloor\ \gamma, a(\gamma), a(\gamma), p''} & \gamma, x, y, p \end{array}$$

commutes even judgementally and $\lfloor \gamma, p'' \rfloor\ \gamma, a(\gamma), a(\gamma), p''$ represents the extension:

$$(\ [\gamma] \xrightarrow{[\lfloor \gamma \rfloor\ \gamma, a(\gamma) : A(\gamma)]} [\gamma, x] \xrightarrow{[\lfloor \gamma \rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)].[\lfloor \gamma \rfloor\ A(\gamma) : \text{Type}]} [\gamma, x, y]\ ).\mathsf{id}_{[\lfloor \gamma \rfloor\ A(\gamma) : \text{Type}]}$$

then $\mathsf{r}^{[\lfloor \gamma \rfloor\ \gamma : \Gamma, a(\gamma) : A(\gamma)]}_{[\lfloor \gamma \rfloor\ A(\gamma) : \text{Type}]}$ is represented by $\lfloor \gamma \rfloor\ \gamma, \mathsf{r}(a(\gamma))$ and we are left to verify that:

$$[\lfloor \gamma \rfloor\ \gamma, p(\gamma) : a(\gamma) = b(\gamma)] = [\lfloor \gamma \rfloor\ \gamma, \mathsf{r}(a(\gamma))].$$

Let us consider a canonical context homotopy equivalence $\langle\ \varphi \mid \psi\ \rangle$ between γ and itself such that $\lfloor \gamma \rfloor\ \alpha(\gamma) : \mathsf{c}(\gamma) = \gamma$ and let $\langle\ \varphi \mid \psi\ \rangle$ be a canonical homotopy equivalence between $A(\gamma)$ and itself relative to $\langle\ \mathsf{c} \mid \mathsf{d}\ \rangle$ and such that $\lfloor \gamma, x : A(\gamma) \rfloor\ \alpha_1(\gamma, x) : \varphi(\gamma, x) = \alpha(\gamma)^* x$ (see Proposition 4.2 and Lemma 4.3). Now, if:

$$\lfloor \gamma \rfloor\ \mathsf{r}_1(\gamma) \equiv \alpha_1(\gamma, a(\gamma)) \bullet a(\alpha(\gamma))^{-1} : \varphi(\gamma, a(\gamma)) = a(\mathsf{c}(\gamma))$$
$$\lfloor \gamma \rfloor\ \mathsf{r}_2(\gamma) \equiv \alpha_1(\gamma, a(\gamma)) \bullet a(\alpha(\gamma))^{-1} \bullet p(\mathsf{c}(\gamma)) : \varphi(\gamma, a(\gamma)) = b(\mathsf{c}(\gamma))$$

then the corresponding $\langle\ \varphi^= \mid \psi^=\ \rangle$ of Lemma 3.5 happens to be a canonical homotopy equivalence between $a(\gamma) = a(\gamma)$ and $a(\gamma) = b(\gamma)$ relative to $\langle\ \mathsf{c} \mid \mathsf{d}\ \rangle$ (see the notion of canonical homotopy equivalence in Subsection 3.6, as well as Remark 3.6) and by propositional groupoidality:

$$\lfloor \gamma \rfloor\ \varphi^=(\gamma, \mathsf{r}(a(\gamma))) = p(\mathsf{c}(\gamma)).$$

Therefore, the diagram:

$$\begin{array}{ccc} \gamma & \xrightarrow{\lfloor \gamma \rfloor\ \gamma, \mathsf{r}(a(\gamma))} & \gamma, p'' \\ {\scriptstyle \mathsf{c}} \downarrow & & \downarrow {\scriptstyle \mathsf{c}, \varphi^=} \\ \gamma & \xrightarrow{\lfloor \gamma \rfloor\ \gamma, p(\gamma)} & \gamma, p' \end{array}$$

commutes propositionally and, by the notion of canonical context homotopy equivalence, the morphism of contexts $\mathsf{c}, \varphi^=$ is the first component of a canonical context homotopy equivalence $\gamma, p'' \to \gamma, p'$ (see Lemma 3.2). Therefore $\lfloor \gamma \rfloor\ \gamma, \mathsf{r}(a(\gamma))$ and $\lfloor \gamma \rfloor\ \gamma, p(\gamma)$ represent the same morphism of semantic contexts i.e. the same arrow of $\mathcal{C}$ and we are done.

- *Compatibility with the substitution.* If we are given a morphism of contexts:

$$[\delta : \Delta] \xrightarrow{[\lfloor \delta \rfloor\ f(\delta) : \Gamma]} [\gamma : \Gamma]$$

(such a representative exists by 3. of Remark 5.8) then the semantic context extension $[\lfloor \delta \rfloor\ f(\delta) : \Gamma].[\lfloor \gamma \rfloor\ A(\gamma) : \text{Type}]$ admits the presentation:

$$[\delta, x' : A(f(\delta))] \xrightarrow{[\lfloor \delta, x' \rfloor\ f(\delta), x']} [\gamma, x : A(\gamma)]$$

and the further semantic context extension:

$$([\lfloor\delta\rfloor\ f(\delta) : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]).[\lfloor\gamma, x : A(\gamma)\rfloor\ A(\gamma) : \textsc{Type}]$$

admits the presentation:

$$[\delta, x', y' : A(f(\delta))] \xrightarrow{[\lfloor\delta,x',y'\rfloor\ f(\delta),x',y']} [\gamma, x, y : A(\gamma)].$$

Hence, by Proposition 4.2, by Corollary 4.10 and by Lemma 4.12, the type:

$$\textsc{tp}(([\lfloor\delta\rfloor\ f(\delta) : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]).[\lfloor\gamma, x : A(\gamma)\rfloor\ A(\gamma) : \textsc{Type}])\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}$$

in context $[\delta, x', y']$ is represented by $\lfloor\delta, x', y'\rfloor\ x' = y'$, which is a representative of $\mathsf{id}_{[\lfloor\delta\rfloor\ A(f(\delta)):\textsc{Type}]}$, and $\lfloor\delta\rfloor\ A(f(\delta)) : \textsc{Type}$ represents the type:

$$(\textsc{tp}[\lfloor\delta\rfloor\ f(\delta) : \Gamma])[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]$$

in context δ, again by Proposition 4.2, by Corollary 4.10 and by Lemma 4.12. We conclude that:

$$\begin{aligned}\textsc{tp}(([\lfloor\delta\rfloor\ f(\delta) : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]).[\lfloor\gamma, x : A(\gamma)\rfloor\ A(\gamma) : \textsc{Type}])\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]} = \\ = \mathsf{id}_{(\textsc{tp}[\lfloor\delta\rfloor\ f(\delta):\Gamma])[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}.\end{aligned}$$

Now, the semantic context extension:

$$(([\lfloor\delta\rfloor\ f(\delta) : \Gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \textsc{Type}]).[\lfloor\gamma, x : A(\gamma)\rfloor\ A(\gamma) : \textsc{Type}]).\mathsf{id}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}$$

admits the presentation:

$$[\delta, x', y', p' : x' = y'] \xrightarrow{[\lfloor\delta,x',y',p'\rfloor\ f(\delta),x',y',p']} [\gamma, x, y, p : x = y]$$

hence we are left to verify that:

$$\begin{array}{ccc} [\delta, x'] & \xrightarrow{[f(\delta),x']} & [\gamma, x] \\ \Big\downarrow{\scriptstyle \mathsf{r}_{[\lfloor\delta\rfloor\ A(f(\delta))]}} & & \Big\downarrow{\scriptstyle \mathsf{r}_{[\lfloor\gamma\rfloor\ A(\gamma)]}} \\ [\delta, x', y', p'] & \xrightarrow{[f(\delta),x',y',p']} & [\gamma, x, y, p] \end{array}$$

commutes in $\mathcal{C}$. By 1. and 4. of Remark 5.8, we are done since the diagram of the representatives:

$$\begin{array}{ccc} \delta, x' & \xrightarrow{f(\delta),x'} & \gamma, x \\ \Big\downarrow{\scriptstyle \delta,x',x',\mathsf{r}(x')} & \equiv & \Big\downarrow{\scriptstyle \gamma,x,x,\mathsf{r}(x)} \\ \delta, x', y', p' & \xrightarrow{f(\delta),x',y',p'} & \gamma, x, y, p \end{array}$$

commutes even judgmentally. □

6.3. **Semantic dependent product and sum types.** Let $[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]$ be a semantic type in semantic context $[\gamma : \Gamma]$ and let:

$$[\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}]$$

be a semantic type in semantic context $[\gamma].[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}] = [\gamma, x]$, where, as usual, we refer to see 1. of Remark 5.10 for the presentation we use for the semantic types. Below, we define the choices of the semantic types and the semantic terms that we need so that $\mathfrak{C}$ is endowed with semantic dependent products and semantic dependent sums respectively.

Π *Formation.* We define the semantic type $\mathbf{\Pi}^{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\text{Type}]}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}$ in context $[\gamma]$ as the semantic type of $\mathfrak{C}$ represented by the type judgement $\lfloor\gamma\rfloor\ \Pi_{x:A(\gamma)}B(\gamma, x) : \text{Type}$.

Introduction. If we are given a semantic term:

$$[\gamma, x] \xrightarrow{[\lfloor\gamma,x\rfloor\ \gamma,x,b(\gamma,x)]} [\gamma, x, y : B(\gamma, x)]$$

(see Remark 6.1 to justify this presentation of the semantic terms) of semantic type:

$$[\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}]$$

we define $\lambda[\lfloor\gamma, x\rfloor\ \gamma, x, b(\gamma, x)]$ to be the semantic term $[\gamma] \to [\gamma, z : \Pi_{x:A(\gamma)}B(\gamma, x)]$ of sematic type $[\lfloor\gamma\rfloor\ \Pi_{x:A(\gamma)}B(\gamma, x) : \text{Type}]$ represented by the morphism of contexts:

$$\lfloor\gamma\rfloor\ \gamma, \lambda x.b(x) : \Pi_{x:A(\gamma)}B(\gamma, x)$$

of PTT.

Elimination. If we are given semantic terms:

$$[\gamma] \xrightarrow{[\lfloor\gamma\rfloor\ \gamma,z(\gamma)]} [\gamma, z] \quad \text{and} \quad [\gamma] \xrightarrow{[\lfloor\gamma\rfloor\ \gamma,a(\gamma)]} [\gamma, x]$$

of semantic type $[\lfloor\gamma\rfloor\ \Pi_{x:A(\gamma)}B(\gamma, x) : \text{Type}]$ and $[\lfloor\gamma\rfloor\ A(\gamma) : \text{Type}]$ respectively, then we define the semantic term:

$$[\gamma] \xrightarrow{\mathsf{ev}^{[\lfloor\gamma\rfloor\ \gamma,a(\gamma)]}_{[\lfloor\gamma\rfloor\ \gamma,z(\gamma)]}} [\gamma, y' : B(\gamma, a(\gamma))]$$

of semantic type:

$$(\text{tp}[\lfloor\gamma\rfloor\ \gamma, a(\gamma)])[\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}] = [\lfloor\gamma\rfloor\ B(\gamma, a(\gamma))]$$

as the one represented by the morphism of contexts:

$$\lfloor\gamma\rfloor\ \gamma, \mathsf{ev}(z(\gamma), a(\gamma)) : B(\gamma, a(\gamma))$$

of PTT (the presentation we use for $(\text{tp}[\lfloor\gamma\rfloor\ \gamma, a(\gamma)])[\lfloor\gamma, x : A(\gamma)\rfloor\ B(\gamma, x) : \text{Type}]$ is justified as usual by Proposition 4.2, by Corollary 4.10 and by Lemma 4.12).

One can prove that:

Proposition 6.3. *With the above choices the properties of* compatibility with the substitution, computation, *and* expansion *of Definition 2.13 are satisfied, hence $\mathfrak{C}$ is equipped with semantic dependent product types.*

Σ *Formation.* We define the semantic type $\mathbf{\Sigma}^{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\text{Type}]}_{[\lfloor\gamma\rfloor\ A(\gamma):\text{Type}]}$ in context $[\gamma]$ as the semantic type of $\mathfrak{C}$ represented by the type judgement $\lfloor\gamma\rfloor\ \Sigma_{x:A(\gamma)}B(\gamma, x) : \text{Type}$.

Introduction. We define the morphism $p^{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\textsc{Type}]}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}$ making the diagram:

$$\begin{array}{ccc}
[\gamma,x,y] & \xrightarrow{p^{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\textsc{Type}]}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}} & [\gamma,u] \\
\Big\downarrow{\scriptstyle P_{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\textsc{Type}]}} & & \Big\downarrow{\scriptstyle P_{\boldsymbol{\Sigma}^{[\lfloor\gamma,x\rfloor\ B(\gamma,x):\textsc{Type}]}_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}}} \\
[\gamma,x] & \xrightarrow{P_{[\lfloor\gamma\rfloor\ A(\gamma):\textsc{Type}]}} & [\gamma]
\end{array}$$

commute (where $u : \Sigma_{x:A(\gamma)}B(\gamma,x) : \textsc{Type}$) as the one represented by the morphism of contexts:

$$\lfloor\gamma,x,y\rfloor\ \gamma,\langle x,y\rangle$$

of PTT.

Elimination. If we are given a semantic type $[\lfloor\gamma,u\rfloor\ C(\gamma,u) : \textsc{Type}]$ in semantic context $[\gamma,u]$ and a semantic term $[\lfloor\gamma,x,y\rfloor\ \gamma,x,y,c(\gamma,x,y) : C(\gamma,\langle x,y\rangle)]$ of semantic type:

$$(\textsc{tp}[\lfloor\gamma,x,y\rfloor\ \gamma,\langle x,y\rangle])[\lfloor\gamma,u\rfloor\ C(\gamma,u) : \textsc{Type}] = [\lfloor\gamma,x,y\rfloor\ C(\gamma,\langle x,y\rangle)]$$

—as usual, use 1. of Remark 5.10 to justify the presentation we use for the semantic types; use Remark 6.1 for the one of the semantic terms; use Proposition 4.2, Corollary 4.10 and Lemma 4.12 for the one of the semantic type $(\textsc{tp}[\lfloor\gamma,x,y\rfloor\ \gamma,\langle x,y\rangle])[\lfloor\gamma,u\rfloor\ C(\gamma,u) : \textsc{Type}]$—then we define the semantic term:

$$[\gamma,u] \xrightarrow{\mathsf{split}_{[\lfloor\gamma,x,y\rfloor\ \gamma,x,y,c(\gamma,x,y):C(\gamma,\langle x,y\rangle)]}} [\gamma,u,c]$$

of semantic type $[\lfloor\gamma,u\rfloor\ C(\gamma,u) : \textsc{Type}]$ as the one represented by the morphism of contexts:

$$\lfloor\gamma,u\rfloor\ \gamma,u,\mathsf{split}(c,\gamma,u) : C(\gamma,u)$$

of PTT.

Again, one can prove that:

Proposition 6.4. *With the above choices the properties of* computation *and* compatibility with the substitution *of Definition 2.15 are satisfied, hence* $\mathfrak{C}$ *is equipped with semantic dependent sum types.*

We might summarise the results of the current section into the following:

Theorem 6.5. *The category with attributes* $\mathfrak{C}$ *is a model of* ETT*.*

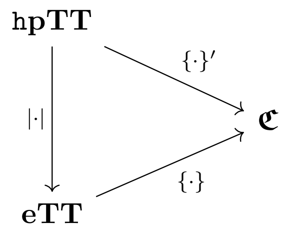
6.4. **Conservativity result.** We remind that hPTT indicates the sub-theory of PTT generated by the atomic h-sets of PTT. In other words, hPTT is the propositional type theory whose contexts are the h-elementary contexts of PTT, whose type judgements are the h-elementary type judgements of PTT and whose term judgements are the term judgements of PTT of an h-elementary type in h-elementary context. We refer to Subsection 2.3 for further details.

We proved that the category with attributes $\mathfrak{C}$ is a model of ETT (Theorem 6.5). By Theorem 2.22 and Remark 2.20 there are:

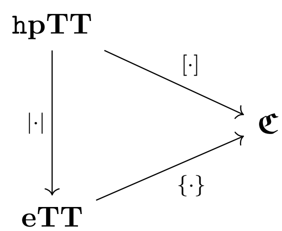
- a unique morphism $\{\cdot\} : \mathbf{eTT} \to \mathfrak{C}$ of ETT;

• a unique morphism $\{\cdot\}' : \mathbf{hpTT} \to \mathfrak{C}$ of hPTT.

The diagram:



where $|\cdot|$ is the canonical interpretation of Remark 2.23, commutes by Remark 2.20 and by Theorem 2.22. However, the quotient mapping $[\cdot]$ defined by the equivalence relation of Section 5 happens to be a morphism $\mathbf{hpTT} \to \mathfrak{C}$ of hPTT. Since $\{\cdot\}' = [\cdot]$ by Theorem 2.22, the diagram:



commutes. Moreover, one can verify that the morphism of semantic types $\mathfrak{C} \to \mathbf{eTT}$ of Remark 5.15 is a morphism of $\mathbf{eTT}$. By Theorem 2.22 it must be a retraction of $\{\cdot\}$, hence $\{\cdot\}$ is injective on semantic contexts, morphisms between them, and semantic types.

Now, let $\gamma : \Gamma$ be an h-elementary context of PTT and let $\lfloor \gamma : \Gamma \rfloor\ T(\gamma) : \textsc{Type}$ be an h-elementary judgement of PTT. Let us suppose that:

$$\lfloor |\gamma : \Gamma| \rfloor_{\mathtt{ext}}\ t(|\gamma : \Gamma|) : |T(\gamma)|$$

in ETT. Here we recall that $|T(\gamma)|$ denotes the type of ETT in context $|\gamma|$ constituting the type judgement $|\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}|$ in ETT (remind that $|\cdot|$ needs to map semantic types in semantic context γ—i.e. h-elementary type judgements of PTT in context γ—to semantic types in semantic context $|\gamma|$—i.e. type judgements of ETT in context $|\gamma|$—), see Remark 2.23 for more details. Hence, in $\mathbf{eTT}$ the judgement $\lfloor |\gamma| \rfloor_{\mathtt{ext}}\ |\gamma|, t(|\gamma|) : |T(\gamma)|$ is a morphism of semantic contexts:

$$|\gamma| \to |\gamma|.|\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}|$$

happening to be a section of the display map $|\gamma|.|\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}| \xrightarrow{P_{|\lfloor \gamma \rfloor\ T(\gamma):\textsc{Type}|}} |\gamma|$. Therefore, being $\{\cdot\}$ a morphism of ETT, we get a section:

$$\begin{aligned}[\gamma] = \{|\gamma|\} \xrightarrow{\{\lfloor |\gamma| \rfloor_{\mathtt{ext}}\ |\gamma|, t(|\gamma|) : |T(\gamma)|\}} \{|\gamma|.|\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}|\} &= \{|\gamma|\}.\{|\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}|\} \\ &= [\gamma].[\lfloor \gamma \rfloor\ T(\gamma) : \textsc{Type}] \\ &= [\gamma : \Gamma, t : T(\gamma)]\end{aligned}$$

of:

$$[\gamma : \Gamma, t : T(\gamma)] \xrightarrow{P_{\{|\lfloor \gamma \rfloor\ T(\gamma):\textsc{Type}|\}} = P_{[\lfloor \gamma \rfloor\ T(\gamma):\textsc{Type}]}} [\gamma]$$

in $\mathfrak{C}$, where the last equality follows by Proposition 4.2 and Corollary 4.10. By Remark 6.1, the arrow $\{\lfloor |\gamma : \Gamma| \rfloor_{\mathtt{ext}}\ |\gamma|, t(|\gamma|) : |T(\gamma)|\}$ happens to be represented by a morphism of contexts:

$$\lfloor \gamma : \Gamma \rfloor\ \gamma : \Gamma, \tilde{t}(\gamma) : T(\gamma)$$

for some judgement $\lfloor \gamma : \Gamma \rfloor\ \tilde{t}(\gamma) : T(\gamma)$. In particular:

$$\begin{aligned}\{\lfloor |\gamma| \rfloor_{\texttt{ext}}\ |\gamma|, |\tilde{t}(\gamma)| : |T(\gamma)|\} &= \{|\lfloor \gamma \rfloor\ \gamma,\ \tilde{t}(\gamma) : T(\gamma)|\} \\ &= [\lfloor \gamma \rfloor\ \gamma,\ \tilde{t}(\gamma) : T(\gamma)] \\ &= \{\lfloor |\gamma| \rfloor_{\texttt{ext}}\ |\gamma|, t(|\gamma|) : |T(\gamma)|\}\end{aligned}$$

and, by injectivity of $\{\cdot\}$ on morphisms of semantic contexts, we conclude that:

$$\lfloor |\gamma| \rfloor_{\texttt{ext}}\ |\tilde{t}(\gamma)| \equiv t(|\gamma|).$$

Hence we have just proven the following:

Theorem 6.6 (Conservativity)**.** *Let* $\gamma : \Gamma$ *be an h-elementary context of* PTT *and let* $\lfloor \gamma : \Gamma \rfloor\ T(\gamma : \Gamma) : \textsc{Type}$ *be an h-elementary type judgement of* PTT. *Whenever* ETT *infers* $\lfloor |\gamma : \Gamma| \rfloor_{\texttt{ext}}\ t(|\gamma : \Gamma|) : |T(\gamma)|$, *then* PTT *infers* $\lfloor \gamma : \Gamma \rfloor\ \tilde{t}(\gamma : \Gamma) : T(\gamma : \Gamma)$ *and* ETT *infers* $\lfloor |\gamma : \Gamma| \rfloor_{\texttt{ext}}\ |\tilde{t}(\gamma : \Gamma)| \equiv t(|\gamma : \Gamma|)$.

In particular, we infer that:

Theorem 6.7. *The theory* ETT *is conservative over the theory* PTT + UIP.

7. Inference rules for dependent type theories

In this section we enumerate the inference rules that we consider in this paper, particularly for what we call *extensional*, *intensional*, and *propositional* type theories with three type constructors: identities, dependent products and dependent sums. We omit the usual structural rules of a strict dependent type theory—that every theory considered in this paper is assumed to satisfy. For an enumeration of the structural rules, see [Str91, Chapter III—context formation, context equality, judgement formation, judgement equality], [Jac99], [Hof97].

7.1. Identity types.

Formation $$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{TYPE}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma) \rfloor\ x = x' : \text{TYPE}}$$

Introduction $$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{TYPE}}{\lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ \mathsf{r}(x) : x = x}$$

Extensionality $$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{TYPE}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ x \equiv x'}$$

Identity proof irrelevance $$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{TYPE}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ p \equiv \mathsf{r}(x)}$$

Figure 1: Extensional identity types

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma) \rfloor\ x = x' : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ \mathsf{r}(x) : x = x}$$

Elimination
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ C(\gamma, x, x', p) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ q(\gamma, x) : C(\gamma, x, x, \mathsf{r}(x))\end{array}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ \mathsf{J}(q, \gamma, x, x', p) : C(\gamma, x, x', p)}$$

Computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ C(\gamma, x, x', p) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ q(\gamma, x) : C(\gamma, x, x, \mathsf{r}(x))\end{array}}{\lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ \mathsf{J}(q, \gamma, x, x, \mathsf{r}(x)) \equiv q(\gamma, x)}$$

Figure 2: Intensional identity types

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma) \rfloor\ x = x' : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ \mathsf{r}(x) : x = x}$$

Elimination
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ C(\gamma, x, x', p) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ q(\gamma, x) : C(\gamma, x, x, \mathsf{r}(x))\end{array}}{\lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ \mathsf{J}(q, \gamma, x, x', p) : C(\gamma, x, x', p)}$$

Prop computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x, x' : A(\gamma);\ p : x = x' \rfloor\ C(\gamma, x, x', p) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ q(\gamma, x) : C(\gamma, x, x, \mathsf{r}(x))\end{array}}{\lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ \mathsf{H}(q, \gamma, x) : \mathsf{J}(q, \gamma, x, x, \mathsf{r}(x)) = q(\gamma, x)}$$

Figure 3: Propositional identity types

7.2. Dependent product types.

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Pi_{x:A(\gamma)} B(\gamma, x) : \textsc{Type}}$$

Elimination
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\, z : \Pi_{x:A(\gamma)} B(\gamma, x);\ x : A(\gamma) \rfloor\ \mathsf{ev}(z, x) : B(\gamma, x)}$$

Introduction
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ y(\gamma, x) : B(\gamma, x)\end{array}}{\lfloor \gamma : \Gamma \rfloor\ \lambda x.y(\gamma, x) : \Pi_{x:A(\gamma)} B(\gamma, x)}$$

Computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ y(\gamma, x) : B(\gamma, x)\end{array}}{\lfloor \gamma : \Gamma \rfloor\ \mathsf{ev}(\lambda x.y(\gamma, x), x) \equiv y(\gamma, x)}$$

Figure 4: Dependent product types

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Pi_{x:A(\gamma)} B(\gamma, x) : \text{Type}}$$

Elimination
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type}}{\lfloor \gamma : \Gamma;\ z : \Pi_{x:A(\gamma)} B(\gamma, x);\ x : A(\gamma) \rfloor\ \mathsf{ev}(z, x) : B(\gamma, x)}$$

Introduction
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ y(\gamma, x) : B(\gamma, x)\end{array}}{\lfloor \gamma : \Gamma \rfloor\ \lambda x.y(\gamma, x) : \Pi_{x:A(\gamma)} B(\gamma, x)}$$

Prop computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ y(\gamma, x) : B(\gamma, x)\end{array}}{\lfloor \gamma : \Gamma \rfloor\ \beta^{\Pi}(y, \gamma, x) : \mathsf{ev}(\lambda x.y(\gamma, x), x) = y(\gamma, x)}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type}}{\begin{array}{c}\lfloor \gamma : \Gamma;\ \ z, z' : \Pi_{x:A(\gamma)} B(\gamma, x);\ p : \Pi_{x:A(\gamma)} \mathsf{ev}(z, x) = \mathsf{ev}(z', x) \rfloor \\ \mathsf{funext}(\gamma, z, z', p) : z = z'\end{array}}$$

Prop computation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type}}{\begin{array}{c}\lfloor \gamma : \Gamma;\ \ z, z' : \Pi_{x:A(\gamma)} B(\gamma, x);\ p : \Pi_{x:A(\gamma)} \mathsf{ev}(z, x) = \mathsf{ev}(z', x) \rfloor \\ \beta_{\mathsf{funext}}(\gamma, z, z', p) : \lambda x.\mathsf{ev}(\mathsf{funext}(\gamma, z, z', p), x) = p\end{array}}$$

Prop expansion
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \text{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \text{Type}}{\begin{array}{c}\lfloor \gamma : \Gamma;\ \ z, z' : \Pi_{x:A(\gamma)} B(\gamma, x);\ q : z = z' \rfloor \\ \eta_{\mathsf{funext}}(\gamma, z, z', q) : q = \mathsf{funext}(\gamma, z, z', \lambda x.\mathsf{ev}(q, x))\end{array}}$$

Figure 5: Propositional dependent product types

7.3. Dependent sum types.

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Sigma_{x:A(\gamma)} B(\gamma, x) : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\, x : A(\gamma), y : B(\gamma, x) \rfloor\ \langle x, y \rangle : \Sigma_{x:A(\gamma)} B(\gamma, x)}$$

Elimination
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ C(\gamma, u) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, x : A(\gamma);\ y : B(\gamma, x) \rfloor\ c(\gamma, x, y) : C(\gamma, \langle x, y \rangle)\end{array}}{\lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \mathsf{split}(c, \gamma, u) : C(\gamma, u)}$$

Computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ C(\gamma, u) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\, x : A(\gamma);\ y : B(\gamma, x) \rfloor\ c(\gamma, x, y) : C(\gamma, \langle x, y \rangle)\end{array}}{\lfloor \gamma : \Gamma;\, x : A(\gamma);\ y : B(\gamma, x) \rfloor\ \mathsf{split}(c, \gamma, \langle x, y \rangle) \equiv c(\gamma, x, y)}$$

Figure 6: Dependent sum types

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma,\ x) : \textsc{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Sigma_{x:A(\gamma)} B(\gamma,\ x) : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma,\ x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x : A(\gamma), y : B(\gamma,\ x) \rfloor\ \langle x, y \rangle : \Sigma_{x:A(\gamma)} B(\gamma,\ x)}$$

Elimination
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma,\ x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma,\ x) \rfloor\ C(\gamma,\ u) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma);\ y : B(\gamma,\ x) \rfloor\ c(\gamma,\ x, y) : C(\gamma,\ \langle x, y \rangle)\end{array}}{\lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma,\ x) \rfloor\ \mathsf{split}(c, \gamma,\ u) : C(\gamma,\ u)}$$

Prop computation
$$\frac{\begin{array}{c}\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma,\ x) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma,\ x) \rfloor\ C(\gamma,\ u) : \textsc{Type} \\ \lfloor \gamma : \Gamma;\ x : A(\gamma);\ y : B(\gamma,\ x) \rfloor\ c(\gamma,\ x, y) : C(\gamma,\ \langle x, y \rangle)\end{array}}{\begin{array}{c}\lfloor \gamma : \Gamma;\ x : A(\gamma);\ y : B(\gamma,\ x) \rfloor \\ \sigma(c, \gamma,\ x, y) : \mathsf{split}(c, \gamma,\ \langle x, y \rangle) = c(\gamma,\ x, y)\end{array}}$$

Figure 7: Propositional dependent sum types

In the next section we recall a characterisations of dependent sum types and propositional sum types *inside* ETT and PTT, respectively. In fact we recall that:

- a dependent type theory with extensional identity types has the rules of Figure 6 if and only if it has the rules of Figure 8;
- a dependent type theory with propositional identity types has the rules of Figure 7 if and only if it has the rules of Figure 9.

For further details, we refer the reader to [Jac99, See84] and to [Spa22], respectively.

7.4. Characterisation of dependent sum types.

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Sigma_{x:A(\gamma)} B(\gamma, x) : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x : A(\gamma), y : B(\gamma, x) \rfloor\ \langle x, y \rangle : \Sigma_{x:A(\gamma)} B(\gamma, x)}$$

Projection
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \pi_1 u : A(\gamma)}$$

$$\lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \pi_2 u : B(\gamma, \pi_1(u))$$

β-reduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ x : A(\gamma);\ y : B(\gamma, x) \rfloor\ \pi_1 \langle x, y \rangle \equiv x}$$

$$\lfloor \gamma : \Gamma;\ x : A(\gamma);\ y : B(\gamma, x) \rfloor\ \pi_2 \langle x, y \rangle \equiv y$$

η-expansion
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\ x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\ u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ u \equiv \langle \pi_1 u, \pi_2 u \rangle}$$

Figure 8: Dependent sum types—negative form

Formation
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma \rfloor\ \Sigma_{x:A(\gamma)} B(\gamma, x) : \textsc{Type}}$$

Introduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\, x : A(\gamma), y : B(\gamma, x) \rfloor\ \langle x, y \rangle : \Sigma_{x:A(\gamma)} B(\gamma, x)}$$

Projection
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\begin{array}{c}\lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \pi_1 u : A(\gamma) \\[1em] \lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \pi_2 u : B(\gamma, \pi_1(u))\end{array}}$$

Prop β-reduction
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\begin{array}{c}\lfloor \gamma : \Gamma;\, x : A(\gamma);\ y : B(\gamma, x) \rfloor\ \beta_1(\gamma, x, y) : \pi_1 \langle x, y \rangle = x \\[1em] \lfloor \gamma : \Gamma;\, x : A(\gamma);\ y : B(\gamma, x) \rfloor\ \beta_2(\gamma, x, y) : \pi_2 \langle x, y \rangle = \beta_1(\gamma, x, y)^* y\end{array}}$$

Prop η-expansion
$$\frac{\lfloor \gamma : \Gamma \rfloor\ A(\gamma) : \textsc{Type} \qquad \lfloor \gamma : \Gamma;\, x : A(\gamma) \rfloor\ B(\gamma, x) : \textsc{Type}}{\lfloor \gamma : \Gamma;\, u : \Sigma_{x:A(\gamma)} B(\gamma, x) \rfloor\ \eta(\gamma, u) : u = \langle \pi_1 u, \pi_2 u \rangle}$$

Figure 9: Propositional dependent sum types—negative form

8. Conclusion

In this paper we compared a propositional dependent type theory to an extensional one, and observed that, despite the non-negligible weakening of the former with respect to the latter, there is actually an interesting family of judgements where the two theories have the same deductive power.

This result was obtained by adapting the argument introduced by Hofmann [Hof96]. In spite of the amount of work in the purely syntactic part of this research, namely in the analysis of the family of canonical equivalences, we underline the central and fundamental role of the soundness property of the semantics induced by the class of categories with attributes. The effect is that, for a given term judgement $t : |T|$ built by ETT, where T is an h-elementary type of PTT, we do not know how a corresponding term $\tilde{t} : T$ of PTT (such that $|\tilde{t}| \equiv t$) is defined. A possible future research direction therefore concerns asking whether there is the possibility of making our argument more constructive from this point of view.

Our argument also applies in the case where PTT and ETT are extended with a (weak) Tarski universe, requiring that it is provably an h-set in PTT and that the types associated with its terms are also provably h-sets in PTT. Clearly, in this setup, the universe cannot be univalent in PTT. To address the issue of accommodating a univalent universe, it would be interesting to explore whether the solution proposed by Winterhalter, Sozeau, and Tabareau [WST19], as applied to two-level type theories, can be used to obtain a corresponding adaptation of our argument.

Our proof, strongly based on canonical equivalences, requires that the family of these be restricted, in such a way that further properties are satisfied. The restriction from general contexts to contexts with h-propositional identities and the one from contexts with h-propositional identities to h-elementary contexts are explicitly applied only at very specific points in the proof. However, they are fundamental for defining—by quotienting—an actual category with attributes, i.e. a model of a dependent type theory. From this perspective, it would then be worthwhile to consider what can be learned by replicating our argument without these restrictions. In detail, one might therefore ask whether the argument can be extended, without restrictions on the family of canonical equivalences, in order to deduce a conservativity result for those *generalised type theories*—with less structural rules—that are modelled by the categorical structure obtained—by quotienting—from the general family of canonical equivalences. More generally, we are interested in looking for other ways of applying Hofmann's argument in order to get similar results or generalised versions of the ones of this paper, depending e.g. on the strictness of the structural rules that we allow in a theory of dependent types.

Acknowledgment

The author thanks his doctoral supervisors Nicola Gambino and Federico Olimpieri for their helpful suggestions during the development of this research. The author is also grateful to (in alphabetic order) Benedikt Ahrens, Ivan di Liberti, Jacopo Emmenegger, Maria Emilia Maietti, Taichi Uemura for useful discussions on the subject. The author would also like to thank the anonymous referees for their useful comments and suggestions. The research presented in this paper was conducted while the author was affiliated with the School of Mathematics of the University of Leeds, United Kingdom.

References

[BCH14] M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. In *19th International Conference on Types for Proofs and Programs*, volume 26 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 107–128. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2014.

[Boc20] R. Bocquet. Coherence of strict equalities in dependent type theories. *arXiv:2010.14166*, 2020. URL: `https://arxiv.org/abs/2010.14166`.

[BW19] S. Boulier and T. Winterhalter. Weak type theory is rather strong. *30th International Conference on Types for Proofs and Programs*, 2019. `https://www.ii.uib.no/~bezem/abstracts/TYPES_2019_paper_18`.

[Car78] J. Cartmell. *Generalised Algebraic Theories and Contextual Categories*. PhD thesis, University of Oxford, 1978.

[CCHM18] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. In *TYPES 2015*, volume 69 of *LIPIcs*. Wadern: Schloss Dagstuhl—Leibniz Zentrum für Informatik, 2018.

[CD13] T. Coquand and N. A. Danielsson. Isomorphism is equality. *Indag. Math. (N.S.)*, 24(4):1105–1120, 2013. `doi:10.1016/j.indag.2013.09.002`.

[CD14] P. Clairambault and P. Dybjer. The biequivalence of locally cartesian closed categories and Martin-Löf type theories. *Mathematical Structures in Computer Science*, 24(6):e240606, 2014. `doi:10.1017/S0960129513000881`.

[CH88] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2):95–120, 1988. `doi:10.1016/0890-5401(88)90005-3`.

[CM24] M. Contente and M. E. Maietti. The compatibility of the minimalist foundation with homotopy type theory. *Theor. Comput. Sci.*, 991:30, 2024. Id/No 114421. `doi:10.1016/j.tcs.2024.114421`.

[Cur93] P.-L. Curien. Substitution up to isomorphism. *Fundamenta Informaticae*, 19(1-2):51–85, 1993.

[Dyb96] P. Dybjer. Internal type theory. In *Types for Proofs and Programs: International Workshop, TYPES'95, Torino, Italy, June 5-8, 1995 Selected Papers*, volume 1158, page 120. Springer Science & Business Media, 1996.

[Gar09] R. Garner. Two-dimensional models of type theory. *Math. Structures Comput. Sci.*, 19(4):687–736, 2009. `doi:10.1017/S0960129509007646`.

[GG08] N. Gambino and R. Garner. The identity type weak factorisation system. *Theoretical Computer Science*, 409(1):94–109, 2008. `doi:10.1016/j.tcs.2008.08.030`.

[Hof95a] M. Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, University of Edinburgh, 1995.

[Hof95b] M. Hofmann. On the interpretation of type theory in locally cartesian closed categories. In L. Pacholski and J. Tiuryn, editors, *Computer Science Logic*, pages 427–441, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[Hof96] M. Hofmann. Conservativity of equality reflection over intensional type theory. In S. Berardi and M. Coppo, editors, *Types for Proofs and Programs*, pages 153–164, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[Hof97] M. Hofmann. *Syntax and semantics of dependent types*, pages 13–54. Springer, London, 1997. `doi:10.1007/978-1-4471-0963-1_2`.

[How80] W. A. Howard. The formulae-as-types notion of construction. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, New York, 1980.

[Jac93] B. Jacobs. Comprehension categories and the semantics of type dependency. *Theoret. Comput. Sci.*, 107(2):169–207, 1993. `doi:10.1016/0304-3975(93)90169-T`.

[Jac99] B. Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

[KL21] K. Kapulkin and P. L. Lumsdaine. Homotopical inverse diagrams in categories with attributes. *Journal of Pure and Applied Algebra*, 225(4):106563, 2021. `doi:10.1016/j.jpaa.2020.106563`.

[KL25] K. Kapulkin and Y. Li. Extensional concepts in intensional type theory, revisited. *Theor. Comput. Sci.*, 1029:29, 2025. Id/No 115051. `doi:10.1016/j.tcs.2024.115051`.

[Mai09] M. E. Maietti. A minimalist two-level foundation for constructive mathematics. *Annals of Pure and Applied Logic*, 160(3):319–354, 2009. Computation and Logic in the Real World: CiE 2007. `doi:10.1016/j.apal.2009.01.006`.

[ML84] P. Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory. Lecture Notes*. Bibliopolis, Naples, 1984. Notes by Giovanni Sambin.

[Mog91] E. Moggi. A category-theoretic account of program modules. *Math. Structures Comput. Sci.*, 1(1):103–139, 1991. doi:10.1017/S0960129500000074.

[MR13] M. E. Maietti and G. Rosolini. Quotient completion for the foundation of constructive mathematics. *Log. Univers.*, 7(3):371–402, 2013. doi:10.1007/s11787-013-0080-2.

[MS25] Maria Emilia Maietti and Pietro Sabelli. Equiconsistency of the minimalist foundation with its classical version. *Ann. Pure Appl. Logic*, 176(2):21, 2025. Id/No 103524. doi:10.1016/j.apal.2024.103524.

[MvG18] S. K. Moss and T. von Glehn. Dialectica models of type theory. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, page 739–748, New York, NY, USA, 2018. Association for Computing Machinery.

[New18] C. Newstead. *Algebraic models of dependent type theory*. PhD thesis, Carnegie Mellon University, 2018. https://www.math.cmu.edu/~cnewstea/thesis-clive-newstead.pdf.

[NPS01] B. Nordström, K. Petersson, and J. Smith. Martin-Löf's type theory. *Handbook of Logic in Computer Science*, 5:1–37, 2001.

[OS25] D. Otten and M. Spadetto. A biequivalence of path categories and axiomatic Martin-Löf type theories. *arXiv:2503.15431*, 2025. URL: https://arxiv.org/abs/2503.15431.

[Our05] N. Oury. Extensionality in the calculus of constructions. In *Theorem proving in higher order logics*, volume 3603 of *Lecture Notes in Comput. Sci.*, pages 278–293. Springer, Berlin, 2005. doi:10.1007/11541868_18.

[Pit00] A. M. Pitts. Categorical logic. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5, pages 39–128. Oxford Univ. Press, 2000.

[RS15] E. Rijke and B. Spitters. Sets in homotopy type theory. *Mathematical Structures in Computer Science*, 25(5):1172–1202, 2015.

[See84] R. A. G. Seely. Locally Cartesian closed categories and type theory. *Math. Proc. Cambridge Philos. Soc.*, 95(1):33–48, 1984. doi:10.1017/S0305004100061284.

[Spa22] M. Spadetto. Towards propositional dependent sums in intensional and propositional dependent type theory. *Preprint*, 2022.

[Spa25] M. Spadetto. A 2-categorical approach to the semantics of dependent type theory with computation axioms. *arXiv:2507.07208*, 2025. URL: https://arxiv.org/abs/2507.07208.

[Str91] T. Streicher. *Semantics of type theory. Correctness, completeness and independence results. With a foreword by Martin Wirsing*. Progress in Theoretical Computer Science. Birkhäuser Boston, Inc., Boston, MA, 1991. doi:10.1007/978-1-4612-0433-6.

[Tay99] P. Taylor. *Practical foundations of mathematics*, volume 59 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1999.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[vdB18] B. van den Berg. Path categories and propositional identity types. *ACM Trans. Comput. Log.*, 19(2):Art. 15, 32, 2018. doi:10.1145/3204492.

[vdBM18] B. van den Berg and I. Moerdijk. Exact completion of path categories and algebraic set theory. Part I: Exact completion of path categories. *J. Pure Appl. Algebra*, 222(10):3137–3181, 2018. doi:10.1016/j.jpaa.2017.11.017.

[Voe15] V. Voevodsky. An experimental library of formalized mathematics based on the univalent foundations. *Mathematical Structures in Computer Science*, 25(5):1278–1294, 2015.

[Win20] T. Winterhalter. *Formalisation and meta-theory of type theory*. PhD thesis, Université de Nantes, 2020.

[WST19] T. Winterhalter, M. Sozeau, and N. Tabareau. Eliminating Reflection from Type Theory. In *CPP 2019 - 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 91–103, Lisbonne, Portugal, January 2019. ACM. doi:10.1145/3293880.3294095.