EXTERNAL UNIVALENCE FOR SECOND-ORDER GENERALIZED ALGEBRAIC THEORIES

RAFAËL BOCQUET

ABSTRACT. Voevodsky's univalence axiom is often motivated as a realization of the equivalence principle; the idea that equivalent mathematical structures satisfy the same properties. Indeed, in Homotopy Type Theory, properties and structures can be transported over type equivalences. However, we may wish to explain the equivalence principle without relying on the univalence axiom. For example, all type formers preserve equivalences in most type theories; thus it should be possible to transport structures over type equivalences even in non-univalent type theories.

We define *external univalence*, a property of type theories (and more general second-order generalized algebraic theories) that captures the preservation of equivalences (or other homotopy relations). This property is defined syntactically, as the existence of identity types on the (syntactically defined) coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF (also called generic model or walking model) of the theory. Semantically, it corresponds to the existence of some left semi-model structure on the category of models of the theory. We give syntactic conditions that can be used to check that a theory satisfies external univalence. We prove external univalence for some theories, such as the first-order generalized algebraic theory of categories, and dependent type theory with any standard choice of type formers and axioms, including identity types, Σ-types, Π-types, universes à la Tarski, the univalence axiom, the Uniqueness of Identity Proofs axiom, etc.

1. INTRODUCTION

The principle of equivalence, also called principle of isomorphism, equivalence-invariance, etc., is the idea that all constructions (in some language or theory) should respect equivalences (for some notion of equivalence associated to the theory). Structures and properties should be transportable over equivalences. Voevodsky's univalence axiom can be seen as an internalization of this principle in the language of type theory. However, univalence is a non-conservative extension of type theory, and incompatible with other useful type theoretic principles, such as Uniqueness of Identity Proofs (UIP). We also wish to achieve transport over equivalent structures in non-univalent type theories.

In some ways, univalence is similar to parametricity. Parametricity captures the preservation of n-ary relations, whereas univalence is related to the preservation of equivalences (which can be seen as binary relations that are functional in both directions). While some theories satisfy internal parametricity, many others only satisfy parametricity externally. External parametricity is a provable metatheoretic property of these theories. In this paper, we introduce *external univalence* a metatheoretic property of theories which captures the preservation of equivalences (or other homotopy relations).

The name "external univalence" corresponds to two ideas. First, as already mentioned, the link between external and internal univalence is somewhat similar to the relationship between external and internal parametricity. Secondly, external univalence is directly related to the other established use of the word "univalence", as found in the notion of univalent category (Ahrens, Kapulkin, and Shulman 2015). Indeed, a theory will satisfy external univalence when its generic model is univalent, for some suitable definition of univalent model of the given theory.

The original motivation for this paper is the author's work (Bocquet 2020) on the conservativity of extensions of type theories by additional definitional equalities. These conservativity results are proven by replacing definitional equalities by transports over equivalences and identifications (elements of the identity type). It is then important to know that equivalences and identifications are preserved by everything in the theory.

We formulate external univalence for any second-order generalized algebraic theory (SOGAT) equipped with the data of homotopy relations on every sort. SOGATs correspond to a class of type theories studied

Date: November 14, 2022.

by Uemura (Uemura 2019; Uemura 2021). The syntax and semantics of most type theories (including type theories with unusual contextual structure, such as cubical type theories and two-level type theories) can all be described using SOGATs. All first-order generalized algebraic theories (GATs, such as the theory of categories) can also be seen as SOGATs. The homotopy relations specify a notion of weak equality on every sort of the theory.

The GAT of categories has three sorts: objects, morphisms and equality between morphisms. The homotopy relation on objects is given by isomorphisms, the homotopy relation on morphisms is given by equality of morphisms, and the homotopy relation on equality of morphisms is trivial. Type theories are usually SOGATs with two sorts: types and terms. The homotopy relations on types and terms can be given by respectively type equivalences and identifications.

We study properties of a SOGAT $\mathcal{T}$ by focusing on its *coclassifying* $(\Sigma, \Pi_{\mathsf{rep}})$-CwF (which contains the *generic model* or *walking model* of $\mathcal{T}$). The coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF is also written $\mathcal{T}$ and is often identified with the theory. The underlying category of $\mathcal{T}$ is equivalent to the category of finitely generated models of $\mathcal{T}$. However, we use a more syntactic definition of $\mathcal{T}$, as the initial model of some two-level type theory. The syntax (i.e. the initial model) of $\mathcal{T}$ embeds faithfully into its coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF; any property of $\mathcal{T}$ has direct consequences on the syntax of the theory. For many theories, the initial model is however too trivial to be interesting; for example, the initial category is empty.

In this setting, we say that a SOGAT $\mathcal{T}$ satisfies external univalence if its coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ can be equipped with (weakly stable and weakly computational) identity types that are compatible with the specified homotopy relations. The elimination principle of these identity types then gives transport over elements that are related by the homotopy relations.

More semantically, we will show in a future article that external univalence is equivalent to the existence of a left semi-model structure on the category of models of $\mathcal{T}$, where the classes of cofibrations, fibrations and weak equivalences are determined by the theory $\mathcal{T}$ and its homotopy relations. In the case of the theory of categories, this semi-model structure is the canonical (or "folk") model structure on **Cat**, while in the case of type theories with identity types, this semi-model structure is the one constructed by Kapulkin and Lumsdaine (2018).

Our main theorem states that external univalence can be proven for a theory by checking some syntactic conditions. Checking these conditions requires providing witnesses of preservation of the homotopy relations by every operation of the theory, together with 1- and 2- dimensional cubical composition and filling operations for the homotopy relations. In the restricted case of theories without equations (e.g. type theories without computation rules), these conditions are actually necessary conditions.

Using this theorem, we show external univalence for:

- the theory of categories, as a minimal example application of the method;
- type theory with identity types and any standard choice of additional type-theoretic structures, such as Π-types, Σ-types, universes à la Tarski (without a coding function), booleans, univalence, UIP, etc.

Depending on the precise algebraic definition of universes, it is not always possible to prove external univalence in the absence of (internal) univalence. A universe comes with a decoding function El, that sends terms of the universe type to types. A coding function is an inverse of the decoding function El, universes with a coding function are also called Coquand universes (Coquand 2013; Coquand 2019). In the absence of a coding function, it is always possible to prove external univalence. Tabareau, Tanter, and Sozeau (2021) give some counter-examples to the preservation of equivalences in the absence of univalence, but they all rely on the use of universes à la Russell, which identify types with terms of the universe.

It should be possible to use our methods to show that other theories (such as the first-order generalized algebraic theory of 2-categories, cubical type theories with or without Glue-types, etc.) also satisfy external univalence.

Example: the theory of categories. We look in more details at external univalence in the setting of the generalized algebraic theory $\mathcal{T}_{\mathbf{Cat}}$ of categories, which is perhaps the simplest theory with non-trivial homotopical content. Since $\mathcal{T}_{\mathbf{Cat}}$ is a *first-order* generalized algebraic theory, its coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}_{\mathbf{Cat}}$ is in fact a coclassifying Σ-CwF. It admits multiple equivalent definitions:

- $\mathcal{T}_{\mathbf{Cat}}$ is the initial category with families equipped with $\mathbf{1}$-types, Σ-types and with an "internal category";
- $\mathcal{T}_{\mathbf{Cat}}$ is the initial model of a type theory with:
  - A type ($\mathbf{ob}\ \mathsf{type}$) of objects;
  - A dependent type $((x, y : \mathbf{ob}) \vdash \mathbf{hom}(x, y)\ \mathsf{type})$ of morphisms;
  - A dependent type $((x, y : \mathbf{ob})\ (f, g : \mathbf{hom}(x, y)) \vdash \mathbf{eqhom}(f, g)\ \mathsf{type})$ of equalities between morphisms;
  - Such that types are closed under $\mathbf{1}$-types and Σ-types;
  - A dependent term $(x : \mathbf{ob}) \vdash \mathbf{id}(x) : \mathbf{hom}(x, x)$;
  - A dependent term $(f : \mathbf{hom}(x, y))\ (g : \mathbf{hom}(y, z)) \vdash \mathbf{comp}(f, g) : \mathbf{hom}(x, z)$;
  - Such that the categorical laws are satisfied:
$$\begin{aligned} &\mathbf{comp}(f, \mathbf{id}) = f, \\ &\mathbf{comp}(\mathbf{id}, f) = f, \\ &\mathbf{comp}(\mathbf{comp}(f, g), h) = \mathbf{comp}(f, \mathbf{comp}(g, h)); \end{aligned}$$
  - And such that the type $\mathbf{eqhom}(f, g)$ is propositional, and inhabited if and only if $f = g$;

  The types and terms of this initial model are respectively the *sorts* and *elements* of the theory of categories.
- The objects of $\mathcal{T}_{\mathbf{Cat}}$ are the categories that are finitely generated by a finite set of objects, a finite collection of morphisms between these objects, and a finite collection of equalities between compositions of these morphisms. The category $\mathcal{T}_{\mathbf{Cat}}$ is a full subcategory of the 1-category $\mathbf{Cat}$.

  The types of $\mathcal{T}_{\mathbf{Cat}}$ over a finitely generated category Γ are the "diagram shapes" over Γ; extensions of Γ by a finite collection of new generating objects, morphisms and equalities. Equivalently, these are the functors into Γ that have a finitely generated domain and are injective-on-objects (i.e. that are cofibrations in the canonical model structure on $\mathbf{Cat}$).

  The terms of a diagram shape A are the actual diagrams of that shape in the category that is finitely generated by Γ.

For example, the context (that can also be seen as a closed record type)
$$\mathsf{Sect} = (x : \mathbf{ob}, y : \mathbf{ob}, r : \mathbf{hom}(x, y), s : \mathbf{hom}(y, x), p : \mathbf{eqhom}(\mathbf{comp}(s, r), \mathbf{id}(y)))$$
is an object of $\mathcal{T}_{\mathbf{Cat}}$. The corresponding finitely generated category is the "walking section"
$$\begin{tikzcd} y \arrow[r, "s"'] \arrow[rr, bend left, "\mathbf{id}(y)"] & x \arrow[r, "r"'] & y . \end{tikzcd}$$

We claim that $\mathcal{T}_{\mathbf{Cat}}$ satisfies external univalence, meaning that the coclassifying Σ-CwF $\mathcal{T}_{\mathbf{Cat}}$ can be equipped with identity types. For any type A, i.e. a diagram shape over Γ, the dependent type $\mathsf{Id}_A(x, y)$ is the diagram shape of isomorphisms between the two copies (x and y) of the diagram A.

For example, the diagram shape $\mathsf{Id}_{\mathsf{Sect}}((x_0, y_0, r_0, s_0, p_0), (x_1, y_1, r_1, s_1, p_1))$ consists of the vertical isomorphisms in the following commutative diagram.
$$\begin{tikzcd} y_0 \arrow[r, "s_0"'] \arrow[rr, bend left, "\mathbf{id}(y_0)"] \arrow[d, "y_2"', "\cong"] & x_0 \arrow[r, "r_0"'] \arrow[d, "x_2"', "\cong"] & y_0 \arrow[d, "y_2"', "\cong"] \\ y_1 \arrow[r, "s_1"] \arrow[rr, bend right, "\mathbf{id}(y_1)"'] & x_1 \arrow[r, "r_1"] & y_1 . \end{tikzcd}$$

The elimination principle of the identity types then tells us that we can transport any diagram extension along such diagram isomorphisms. For instance, if we know that s_0 and r_0 are actually inverses in the above setting, we can transport this fact over the diagram isomorphism to obtain that s_1 and r_1 are also inverses.

Concretely, we can form a dependent type P over Sect, with $P(x,y,r,s) = \mathbf{eqhom}(\mathbf{comp}(r,s), \mathbf{id}(x))$. If we have any term of type $P(x_0,y_0,r_0,s_0,p_0)$, we obtain an element of type $P(x_1,y_1,r_1,s_1,p_1)$ by transport.

Note that we do not include any type of "equality between objects". Indeed, equalities between objects cannot be transported over diagram isomorphisms.

Example: Dependent type theories. We also describe what external univalence entails for a dependent type theory $\mathcal{T}$ with identity types, universes à la Tarski and any choice of standard type formers (Π-types, Σ-types, inductive types, etc.).

The coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF of $\mathcal{T}$ can be described as the initial model of a two-level type theory with:

- For every universe level i, we have:
  - An outer type $(\mathbf{ity}_i\ \mathsf{type})$ of inner types of level i.
  - A dependent outer type $(A : \mathbf{ity}_i \vdash \mathbf{itm}_i(A)\ \mathsf{type})$ of inner terms.
  - An inner type $(\boldsymbol{U}_i : \mathbf{ity}_{i+1})$ for the universe of i-small inner types and a dependent inner type $(A : \mathbf{itm}_{i+1}(\boldsymbol{U}_i) \vdash \mathbf{El}(A) : \mathbf{iTy}_i)$ for its decoding function.
- The inner types and terms are closed under the operations of the dependent type theory $\mathcal{T}$, including identity types $\mathbf{iId}$, $\mathbf{irefl}$, etc.
- The outer types are closed under $\mathbf{1}$- and Σ-types.
- The outer types are closed under Π-types with arities in inner terms. This means that we have a type forming operation

$$\frac{A : \mathbf{ity}_i \qquad a : \mathbf{itm}_i(A) \vdash B(a)\ \mathsf{type}}{\Pi(A,B)\ \mathsf{type}}$$

  such that terms of type $\Pi(A,B)$ correspond bijectively to dependent terms $(a : \mathbf{itm}_i(A) \vdash b : B(a))$.

The underlying category of $\mathcal{T}$ is equivalent to the category of all finitely generated contextual models of $\mathcal{T}$.

The model $\mathcal{T}$ does not coincide with the initial model $\mathbf{0}_{\mathcal{T}}$ of $\mathcal{T}$; but there is a faithful embedding $\mathbf{0}_{\mathcal{T}} \to \mathcal{T}$, so that anything constructed in $\mathcal{T}$ is also valid in the syntax $\mathbf{0}_{\mathcal{T}}$.

In that setting, external univalence for $\mathcal{T}$ says that the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ is equipped with (weakly stable) identity types with $\mathsf{Id}_{\mathbf{ity}_i}(A,B) \simeq \mathrm{iEquiv}(A,B)$ and $\mathsf{Id}_{\mathbf{itm}_i(A)}(x,y) \simeq \mathbf{itm}_i(\mathbf{iId}_A(x,y))$, where $\mathrm{iEquiv}(A,B)$ is the outer type of inner equivalences between the inner types A and B.

A closed dependent inner type $(A : \mathbf{ity}_i \vdash P(A) : \mathbf{ity}_i)$ in the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ is exactly a type expression that depends on a type variable A. If $\mathcal{T}$ satisfies external univalence, we know that any such P perserves equivalences. Indeed, P has an action on paths:

$$(A, B : \mathbf{ity}_i), E : \mathsf{Id}_{\mathbf{ity}_i}(A,B) \vdash \mathsf{ap}(P,E) : \mathsf{Id}_{\mathbf{ity}_i}(P(A),P(B)).$$

By external univalence, this is equivalent to an action of P on equivalences:

$$(A, B : \mathbf{ity}_i), E : \mathrm{iEquiv}(A,B) \vdash \mathsf{ap}(P,E) : \mathrm{iEquiv}(P(A),P(B)).$$

Furthermore, this action of P on equivalences preserves composition of equivalences and the whole ∞-groupoid structure of types.

As an example, we can show how to transport the commutativity of addition from a type $\mathbb{N}_{\mathsf{un}}$ of unary natural numbers to a type $\mathbb{N}_{\mathsf{bin}}$ of binary natural numbers. We consider the following dependent type:

$$\begin{aligned} P &: (N : \mathbf{ity}_0) \times (\mathsf{plus} : \mathbf{itm}(N) \to \mathbf{itm}(N) \to \mathbf{itm}(N)) \to \mathbf{ity}_0, \\ P(N, \mathsf{plus}) &\triangleq \forall n\ m \to \mathbf{iId}_N(\mathsf{plus}(n,m), \mathsf{plus}(m,n)). \end{aligned}$$

We have an identification E between $(\mathbb{N}_{\mathsf{un}}, +_{\mathsf{un}})$ and $(\mathbb{N}_{\mathsf{bin}}, +_{\mathsf{bin}})$ in the outer type $(N : \mathbf{ity}_0) \times (\mathsf{plus} : \mathbf{itm}(N) \to \mathbf{itm}(N) \to \mathbf{itm}(N))$. By external univalence and function extensionality in $\mathcal{T}$, this identification consists of an equivalence between $\mathbb{N}_{\mathsf{un}}$ and $\mathbb{N}_{\mathsf{bin}}$ along with a proof that it is compatible with $+_{\mathsf{un}}$ and $+_{\mathsf{bin}}$. We can then use the action on paths of P to obtain an identification $\mathsf{ap}(P,E)$ between $P(\mathbb{N}_{\mathsf{un}}, +_{\mathsf{un}})$ and $P(\mathbb{N}_{\mathsf{bin}}, +_{\mathsf{bin}})$. By external univalence, we can also see $\mathsf{ap}(P,E)$ as an equivalence between $P(\mathbb{N}_{\mathsf{un}}, +_{\mathsf{un}})$ and $P(\mathbb{N}_{\mathsf{bin}}, +_{\mathsf{bin}})$. Now given any term of type $P(\mathbb{N}_{\mathsf{un}}, +_{\mathsf{un}})$, we can apply the equivalence to obtain a term of type $P(\mathbb{N}_{\mathsf{bin}}, +_{\mathsf{bin}})$, i.e. a proof of commutativity for the binary natural numbers.

Note that if $\mathcal{T}$ has Coquand universes instead, given by inverses $(A : \mathbf{ity}_i \vdash \mathbf{c}(A) : \mathbf{itm}_{i+1}(\boldsymbol{U}_i))$ of the coding functions $\mathbf{El}$, then external univalence implies internal univalence. Indeed the action of $\mathbf{c}$ on paths is

$$(A, B : \mathbf{ity}_i), E : \mathsf{Id}_{\mathbf{ity}_i}(A, B) \vdash \mathsf{ap}(\mathbf{c}, E) : \mathsf{Id}_{\mathbf{itm}_{i+1}(\boldsymbol{U}_i)}(\mathbf{c}(A), \mathbf{c}(B)).$$

By external univalence, this is equivalent to

$$(A, B : \mathbf{ity}_i), E : \mathsf{iEquiv}(A, B) \vdash \mathsf{ap}(\mathbf{c}, E) : \mathbf{itm}_{i+1}(\mathbf{iId}_{\boldsymbol{U}_i}(\mathbf{c}(A), \mathbf{c}(B))),$$

i.e. to the fact that any equivalence can be turned into an identification between elements of the universe $\boldsymbol{U}_i$.

Related work.

Relational parametricity and the Identity Extension Lemma. Reynolds' relational parametricity (Reynolds 1983) provides an interpretation of the types of System F as binary relations for any given mapping of the type variables to relations. A crucial property of Reynolds' model is the Identity Extension Lemma, which states that whenever all type variables are mapped to identity relations, the interpretation of any type is also the identity relation.

For dependent type theories, constructing models that satisfy the Identity Extension Lemma is generally challenging. Atkey, Ghani, and Johann (2014) show that the Identity Extension Lemma can be motivated by the use of reflexive graphs in the construction of relationally parametric models. They also construct a relationally parametric model of dependent types in reflexive graphs. In that model, the universe of small types is interpreted as a universe of discrete and proof-irrelevant reflexive graphs.

Although the general setting differs, external univalence seems to be related to the Identity Extension Lemma, as our goal is to interpret every type as a (type-valued) relation that is also an identity type. We also almost use reflexive graphs in our constructions, except that we have to replace the diagram shape of reflexive graphs by an inverse diagram shape (see section 6).

Univalent Parametricity. Tabareau, Tanter, and Sozeau (2021) give a univalent parametricity translation for a type theory with the univalence axiom. The univalence axiom is needed in their translation of the universes. This translation allows for the transport of proofs and structures over equivalences. In many instances, the transport is effective, meaning that the output term does not actually rely on the univalence axiom. In these cases, the translated terms can be used even in non-univalent type theories.

Tabareau et al. implemented this univalent parametricity using the typeclass mechanism of Coq. Ringer et al. (2019) also implemented a related transformation as a Coq plugin.

In our work, we show that the transport of structures over equivalences can be achieved even for non-univalent type theories, if their universes do not have a coding function. Our constructions involve a homotopical inverse diagram model that is closely related to the univalent parametricity translation of Tabareau et al.

We do not provide any algorithmic implementation of our results. However, we work in a constructive metatheory, it is in principle possible to extract an algorithm from our proofs. Furthermore most of our constructions involve syntactic manipulations that should be directly implementable.

Semi-model structures on categories of models of type theories. Kapulkin and Lumsdaine (2018) construct left semi-model structures on the categories of models of type theories with identity types, Σ-types, and (optionally) Π-types with function extensionality. The properties of these semi-model structures can be used to transport structures over equivalences. They prove the existence of the semi-model structures using several homotopical inverse diagram models. For this purpose, Kapulkin and Lumsdaine (2021) have constructed homotopical inverse diagram models over arbitrary homotopical inverse categories. Note that closely related homotopical gluing models had been constructed before by Shulman (2015). Isaev (2017) has also constructed some model structures on categories of models of type theories with an interval.

We will show in another paper that a SOGAT equipped with homotopy relations satisfies external univalence if and only if its category of models is a left semi-model category, for classes of trivial cofibrations, cofibrations and weak equivalences that are derived from the SOGAT and the chosen homotopy relations. Thus our results will yield an alternative proof of the results of Kapulkin and Lumsdaine. Since we prove external univalence for a large class of type theories, we will also obtain left semi-model structures for a large class of type theories.

Computing with univalence. Proving external univalence for a theory essentially involves providing a computational explanation of univalence in a very restricted setting: the outer layer of the coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF of the theory. It has only Σ-types, some Π-types, and some base types. In particular, there is no universe classifying the outer types, so univalence cannot be iterated. As a consequence, giving a computational explanation of external univalence is much simpler than for internal univalence.

Nevertheless, there are some similarities between our setting and computation with internal univalence. Some of our constructions are reminiscent of the cubical type theory without an interval of Altenkirch and Kaposi (2015), which was an early attempt at providing a computation interpretation of internal univalence.

Principle of equivalence. Makkai's Principle of Isomorphism (Makkai 1998) is the idea that "All grammatically correct properties of objects of a fixed category are to be invariant under isomorphism." These ideas were formally developed in the framework of First-Order Logic with Dependent Sorts (Makkai 1995). There was also prior work by Freyd (1976) and Blanc (1978), showing that first-order categorical statements can be transported over equivalences of categories, as long as they do not mention equalities between objects. In a recent talk, Henry (2020) has explained the relationship between these ideas and homotopy theory.

Ahrens, North, et al. (2020) have revisited FOLDS in a univalent setting, and give a generic definition of "indiscernability" for any FOLDS-signature. FOLDS-signatures can be identified with first-order generalized algebraic theories without operations. It would be interesting to investigate whether indiscernabilities are homotopy relations that always satisfy external univalence in our setting.

∞-type theories. We expressed external univalence using the structure of identity types on the coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF of a SOGAT $\mathcal{T}$. This coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF then has the structure of a model of type theory with Σ-types, (weakly stable) identity types and some Π-types. In line with internal language conjectures (Kapulkin and Lumsdaine 2018; Kapulkin and Szumiło 2017), which assert that models of type theories with identity types and other structures are the internal languages of structured ∞-categories, the coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF ought to be the internal language of some ∞-category with representable maps.

Nguyen and Uemura (2022) have used a precise definition of ∞-categories with representable maps as a notion of ∞-type theory. Such an ∞-type theory has an ∞-category of models; in a model all substitution laws and computation rules only hold up to homotopy. They have also established some coherence theorems that compare some ∞-type theories with some 1-type theories.

Our results provide a way to work with objects that are morally ∞-type theories, without relying on any simplicial presentation of ∞-categories. Instead we morally use a type-theoretic definition of (structured) ∞-categories, originally inspired by Brunerie's type-theoretic definition of ∞-groupoids (Brunerie 2016, Appendix B).

2. Background

We work in a constructive metatheory.

2.1. Notations.

We use different relation symbols for the different notions of identifications that occur in this paper. We reserve the use of $(\sim)$ for homotopy relations associated to a theory (see definition 4.1). The symbol $(\simeq)$ is used for equivalences between types and identifications (terms of an identity type). Isomorphisms are denoted by the symbol $(\cong)$.

2.2. Factorization systems.

We recall some basic results on (both weak and orthogonal) factorization systems over locally finitely presentable categories. We omit all proofs. Details on locally presentable categories can be found in the standard reference book by Adamek and Rosicky (1994). A general introduction to factorization systems can be found in notes by Riehl (2008).

We fix a locally finitely presentable category $\mathbf{C}$.

Definition 2.1. Let $l : A \to B$ and $r : X \to Y$ be two maps in $\mathbf{C}$. We say that l has the **left lifting property** with respect to r, or that r has the **right lifting property** with respect to l if for any square (lifting problem) of the form

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

there exists a diagonal map $h : B \to X$ such that $h \circ l = f$ and $g = r \circ h$. In that case we write $l \boxslash r$.

We say that l has the **unique left lifting property** with respect to r, when the diagonal filler h is unique. This is also denoted by $l \perp r$ ⌟

Proposition 2.2. *Given $l : A \to B$ and $r : X \to Y$, we have $l \perp r$ if and only if $l \boxslash r$ and $\nabla_f \boxslash r$ where $\nabla_f : B +_A B \to B$ is the codiagonal of f.* □

Definition 2.3. A **weak factorization system** consists of two classes $\mathcal{L}$ and $\mathcal{R}$ of maps of $\mathbf{C}$, such that

$$\mathcal{L} = \{l : A \to B \mid \forall r : X \to Y, l \boxslash r\},$$
$$\mathcal{R} = \{r : X \to Y \mid \forall l : A \to B, l \boxslash r\},$$

and such that every map can be factored as a map in $\mathcal{L}$ followed by a map in $\mathcal{R}$. ⌟

Definition 2.4. An **orthogonal factorization system** consists of two classes $\mathcal{L}$ and $\mathcal{R}$ of maps of $\mathbf{C}$, such that

$$\mathcal{L} = \{l : A \to B \mid \forall r : X \to Y, l \perp r\},$$
$$\mathcal{R} = \{r : X \to Y \mid \forall l : A \to B, l \perp r\},$$

and such that every map can be factored as a map in $\mathcal{L}$ followed by a map in $\mathcal{R}$. ⌟

Proposition 2.5. *Any orthogonal factorization system is also a weak factorization system. Conversely, a weak factorization system is an orthogonal factorization system if and only if for every $f \in \mathcal{L}$, $\nabla_f \in \mathcal{L}$.* □

We fix a set $\mathcal{X}$ of maps in $\mathbf{C}$.

Definition 2.6. An **$\mathcal{X}$-cellular map** is a sequential composition of pushouts of coproducts of maps in $\mathcal{X}$. A **$\mathcal{X}$-cellular complex** is an object A of $\mathbf{C}$ such that the unique map $\mathbf{0}_{\mathbf{C}} \to A$ is an $\mathcal{X}$-cellular map.

A **finite $\mathcal{X}$-cellular map** is a finite composition of pushouts of maps in $\mathcal{X}$. A **finite $\mathcal{X}$-cellular complex** is an object A of $\mathbf{C}$ such that the unique map $\mathbf{0}_{\mathbf{C}} \to A$ is a finite $\mathcal{X}$-cellular map. ⌟

We see $\mathcal{X}$-cellularity as additional structure on the maps of $\mathbf{C}$. The cellular maps are usually defined as arbitrary transfinite compositions of pushouts of coproducts of maps in $\mathcal{X}$; but since $\mathbf{C}$ is locally *finitely* presentable, it suffices to consider sequential compositions.

Lemma 2.7 (Small object argument)**.** *There is a weak factorization system on $\mathbf{C}$, said to be cofibrantly generated by $\mathcal{X}$. The right class of maps consists of maps with the right lifting property with respect to every map in $\mathcal{X}$. The maps in the left class are the retracts of $\mathcal{X}$-cellular maps. Furthermore, every map in $\mathbf{C}$ factors as a $\mathcal{X}$-cellular map followed by a map in the right class.* □

Lemma 2.8 (Small object argument for orthogonal factoriation systems)**.** *There is an orthogonal factorization system on $\mathbf{C}$, generated by $\mathcal{X}$. The maps in the right class are the maps with the unique right lifting property with respect to every map in $\mathcal{X}$. As a weak factorization system, it is cofibrantly generated by*

$$\mathcal{X} \cup \{\nabla_f \mid f \in \mathcal{X}\}.$$ □

2.3. Internal language of presheaf categories.

We frequently use the type-theoretic internal languages of presheaf categories throughout this paper.

We use $\mathbf{Psh}(\mathcal{C})$ to refer to the presheaf topos over $\mathcal{C}$; it is a model of extensional type theory with a hierarchy of universes closed under many type-theoretic structures, including dependent products, dependent sums, extensional equality types, quotient types, etc.

We use $\widehat{\mathcal{C}}$ to refer to the presheaf category over $\mathcal{C}$. It could be the underlying category of the topos $\mathbf{Psh}(\mathcal{C})$, but we typically assume that $\widehat{\mathcal{C}}$ lives in a smaller universe than $\mathbf{Psh}(\mathcal{C})$.

The types of $\mathbf{Psh}(\mathcal{C})$ are the dependent presheaves; a dependent presheaf Y over a presheaf X is equivalently a presheaf over the category of elements $\int_{\mathcal{C}} X$.

The universes of $\mathbf{Psh}(\mathcal{C})$ are the Hofmann-Streicher universes; they classify the (i-small) dependent presheaves. We denote them by $\mathcal{U}_i$, or just $\mathcal{U}$.

We often need to reason externally with objects that were defined in the internal language. In that case, we borrow the following notations from crisp type theory (Shulman 2017). If X is a presheaf over $\mathcal{C}$, i.e. a type of $\mathbf{Psh}(\mathcal{C})$ over the empty context of $\mathbf{Psh}(\mathcal{C})$, we write $x :: X$ to indicate that x is a *global* element of X. When the category $\mathcal{C}$ has a terminal object $\diamond$, this means that x is an element of $X_\diamond$. In particular, if $x :: \mathbf{y}(\Gamma) \to X$, then x is a global element of the exponential presheaf $(\mathbf{y}(\Gamma) \to X)$, or equivalently an element of X_Γ by the Yoneda lemma. We leave implicit such uses of the Yoneda lemma. Conversely, whenever we have a global element x of X, we can use it in the internal language of $\mathbf{Psh}(\mathcal{C})$ wherever an element of X would be expected.

2.3.1. *Local representability.* We recall the notion of locally representable dependent presheaves, which is used to model context extensions.

Definition 2.9. A dependent presheaf Y over a presheaf X is **locally representable** when for every element $x :: \mathbf{y}(\Gamma) \to X$, the restricted presheaf

$$\begin{aligned} Y_{|x} &: (\mathcal{C}/\Gamma)^{\mathsf{op}} \to \mathbf{Set}, \\ Y_{|x}(\rho : \Delta \to \Gamma) &\triangleq Y_\Delta(x[\rho]). \end{aligned}$$

is representable.

Its representing object consists of an extended context $\Gamma.Y_{|x}$ along with an isomorphism

$$\langle \boldsymbol{p}, \boldsymbol{q} \rangle : \mathbf{y}(\Gamma.Y_{|x}) \cong (\gamma : \mathbf{y}(\Gamma)) \times Y(x(\gamma)).$$

We will often denote the extended context by $(\gamma : \Gamma).Y(x(\gamma))$ and implicitly coerce through the isomorphism above. ⌟

A dependent presheaf Y is locally representable if and only if the corresponding total natural transformation $\Sigma_X Y \to X$ is a representable natural transformation (Awodey 2018).

There is a universe $\mathcal{U}_{\mathsf{rep}}$ classifying the locally representable dependent presheaves in $\mathbf{Psh}(\mathcal{C})$, see for instance (Streicher 2014) for a construction.

2.4. **Type-theoretic structures over internal families.** We now work internally to a presheaf topos $\mathbf{Psh}(\mathcal{C})$.

2.4.1. *Internal families.*

Definition 2.10. A **family** is a pair $(\mathsf{Ty}, \mathsf{Tm})$, where $\mathsf{Ty} : \mathcal{U}$ and $\mathsf{Tm} : \mathsf{Ty} \to \mathcal{U}$. It is said to have **representable elements** when $\mathsf{Tm}(A)$ is locally representable for any type A, i.e. when $\mathsf{Tm} : \mathsf{Ty} \to \mathcal{U}_{\mathsf{rep}}$. ⌟

The elements of Ty are often called types, and the elements of Tm are called terms.

Definition 2.11. A **restriction** $\mathsf{Ty}' \to \mathsf{Ty}$ of a family $(\mathsf{Ty}, \mathsf{Tm})$ consists of a presheaf $\mathsf{Ty}' : \mathcal{U}$ along with a map $\iota : \mathsf{Ty}' \to \mathsf{Ty}$. It induces a **restricted family** $(\mathsf{Ty}', \mathsf{Tm}')$, with $\mathsf{Tm}'(A) = \mathsf{Tm}(\iota(A))$.

A **subfamily** $\mathsf{Ty}' \hookrightarrow \mathsf{Ty}$ is a restriction that is also a monomorphism. ⌟

We will often leave ι implicit, especially when it is a monomorphism.

2.4.2. *Basic type-theoretic structures.*

Definition 2.12. A $\mathbf{1}$-type structure over a family $(\mathsf{Ty}, \mathsf{Tm})$ consists of a type

$$\mathbf{1} : \mathsf{Ty}$$

along with an isomorphism

$$\mathsf{Tm}(\mathbf{1}) \cong \{\mathsf{tt}\}.$$ ⌟

Definition 2.13. A Σ-type structure over a family $(\mathsf{Ty}, \mathsf{Tm})$ consists of an operation

$$\Sigma : (A : \mathsf{Ty})(B : \mathsf{Tm}(A) \to \mathsf{Ty}) \to \mathsf{Ty}$$

along with an isomorphism

$$\mathsf{Tm}(\Sigma(A, B)) \cong ((a : \mathsf{Tm}(A)) \times (b : \mathsf{Tm}(B(a)))). \qquad \lrcorner$$

Definition 2.14. A Π-type structure over a family $(\mathsf{Ty}, \mathsf{Tm})$ consists of an operation

$$\Pi : (A : \mathsf{Ty})(B : \mathsf{Tm}(A) \to \mathsf{Ty}) \to \mathsf{Ty}$$

along with an isomorphism

$$\mathsf{Tm}(\Pi(A, B)) \cong ((a : \mathsf{Tm}(A)) \to \mathsf{Tm}(B(a))). \qquad \lrcorner$$

We will implicitly coerce through these isomorphisms.

2.4.3. *First-order* Π*-types.* We also need to consider a restriction of Π-types that will be used to describe the binders of type theories.

Definition 2.15. The structure of Π**-types** in a family $(\mathsf{Ty}, \mathsf{Tm})$ **with arities** in a family $(\mathsf{Ty}', \mathsf{Tm}')$ consists of an operation

$$\Pi : (A : \mathsf{Ty}')(B : \mathsf{Tm}'(A) \to \mathsf{Ty}) \to \mathsf{Ty}$$

along with an isomorphism

$$\mathsf{Tm}(\Pi(A, B)) \cong ((a : \mathsf{Tm}'(A)) \to \mathsf{Tm}(B(a))). \qquad \lrcorner$$

Definition 2.16. The structure of **first-order** Π**-types** in a family $(\mathsf{Ty}, \mathsf{Tm})$ consists of a restricted family $\mathsf{RepTy} \to \mathsf{Ty}$, along with Π-types in $(\mathsf{Ty}, \mathsf{Tm})$ with arities in RepTy. $\lrcorner$

The intuition here is that Ty is the family of first-order types, while RepTy is its restricted family of zeroth-order types. The domain of a first-order Π-type has to be a zeroth-order Π-type. Elements of RepTy will also be called *representable types*, since they will typically be interpreted as locally representable dependent presheaves. We sometimes use Π_{rep} to refer to the first-order Π-types.

We use these first-order Π-types in the definition of second-order generalized algebraic theories; in a second-order theory, the domain of an operation can be any first-order type.

Example 2.17. A presheaf topos $\mathbf{Psh}(\mathcal{C})$ is equipped with first-order Π-types, where the representable types are the locally representable dependent presheaves. The first-order Π-types could be defined to be the usual Π-types of the presheaf topos, but there is also an alternative definition that relies on the local representability of the domain. Indeed, if X is a presheaf, Y is a dependent presheaf over X and Z is a dependent presheaf over $\Sigma_X Y$, we can pose

$$\Pi(Y, Z)_\Gamma(x) \triangleq Z_{(\gamma:\Gamma).Y(x(\gamma))}(\lambda(\gamma, -) \mapsto x(\gamma)).$$

In the simply-typed case, this was first observed by Hofmann (1999).

Because the two definitions satisfy the same universal property, they are interchangeable. However the alternative definition gives a first-order algebraic presentation of the categories of models of algebraic theories with binders, ensuring that the category of models is locally finitely presentable and the existence of initial models. $\lrcorner$

2.4.4. *Telescopes.* Given any family $(\mathsf{Ty}, \mathsf{Tm})$, we can consider the family $(\mathsf{Ty}^\star, \mathsf{Tm}^\star)$ of *telescopes*; the notation $\mathsf{Ty}^\star$ is inspired from the notation $A^\star$ for the set of lists of elements of a set A. The elements of $\mathsf{Ty}^\star$ are finite dependent sequences

$$(A_1 : \mathsf{Ty}, A_2 : \mathsf{Tm}(A_1) \to \mathsf{Ty}, A_3 : (a_1 : \mathsf{Tm}(A_1)) \to (a_2 : \mathsf{Tm}(A_2(a_1))) \to \mathsf{Ty}, \dots)$$

of types, and elements of $\mathsf{Tm}^\star(A)$ are sequences

$$(a_1 : \mathsf{Tm}(A_1), a_2 : \mathsf{Tm}(A_2(a_1)), a_3 : \mathsf{Tm}(A_3(a_1, a_2)), \dots)$$

of terms of the types of the sequence A.

Definition 2.18. The family $(\mathsf{Ty}^\star, \mathsf{Tm}^\star)$ is defined by induction-recursion as follows:

$$\begin{aligned}
&\mathsf{Ty}^\star && : \mathcal{U}, \\
&\mathsf{Tm}^\star && : \mathsf{Ty}^\star \to \mathcal{U}, \\
&\diamond && : \mathsf{Ty}^\star, \\
&\mathsf{Tm}^\star(\diamond) && \triangleq \top, \\
&\_.\_ && : (A : \mathsf{Ty}^\star) \to (\mathsf{Tm}^\star(A) \to \mathsf{Ty}^\star) \to \mathsf{Ty}^\star, \\
&\mathsf{Tm}^\star(A.B) && \triangleq (a : \mathsf{Tm}^\star(A)) \times \mathsf{Tm}(B(a)).
\end{aligned}$$
⌟

The family of telescopes can be equipped with (strictly associative and unital) Σ-types, given by concatenation of the sequences of types. When the base family $(\mathsf{Ty}, \mathsf{Tm})$ has Σ-types, there is a family morphism $\mathsf{Ty}^\star \to \mathsf{Ty}$ that interprets telescopes as (either left-nested or right-nested) iterated Σ-types.

2.5. **Categories with Families.** We now return to an external setting. The internal notions of type-theoretic structures yield external notions of models equipped with these type-theoretic structures. More precisely, these models are categories with families (CwFs,Dybjer 1995; Castellan, Clairambault, and Dybjer 2019).

Note that for most of this paper, CwFs are not directly used as the notion of model of type theory, but rather as worlds in which the notion of model of type theory can be interpreted. In other words, they do not correspond to the object theories we are interested in, but rather to logical frameworks in which the object theories can be specified and interpreted. Accordingly, while we study arbitrary object theories, the CwFs will only be equipped with a handful of structures (Σ-types, (first-order) Π-types, and some identity types). These correspond (Clairambault and Dybjer 2014) to well-known classes of structured categories, such as clans, finitely complete categories, representable map categories, locally cartesian closed categories, etc.

Definition 2.19. A **category with families** (CwF) $\mathcal{C}$ is a category, equipped with a terminal object, along with a global family $(\mathsf{Ty}_{\mathcal{C}}, \mathsf{Tm}_{\mathcal{C}})$ with representable elements in $\mathbf{Psh}(\mathcal{C})$. ⌟

We have a locally finitely presentable 1-category **CwF** of CwFs and strict CwF morphisms.

Definition 2.20. A Σ-CwF is a CwF whose family is equipped with **1**- and Σ- types. ⌟

We write $\mathbf{CwF}_\Sigma$ for the 1-category of Σ-CwFs.

Definition 2.21. A $(\Sigma, \Pi_{\mathsf{rep}})$-CwF is a CwF equipped with:

- A restriction $\mathsf{RepTy} \to \mathsf{Ty}$ inducing a family of **representable types** (or first-order types).
- First-order Π-types with respect to $\mathsf{RepTy} \to \mathsf{Ty}$.
- Along with **1**- and Σ- type structures over the families Ty and RepTy. They do not have to be strictly preserved by $\mathsf{RepTy} \to \mathsf{Ty}$ (but they are automatically preserved up to isomorphism). ⌟

We write $\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}}}$ for the 1-category of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs and strict morphisms.

Example 2.22. Any presheaf category $\widehat{\mathcal{C}}$ is equipped with the structure of a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF where:

- The types are the dependent presheaves.
- The representable types are the locally representable dependent presheaves.
- The first-order Π-types are defined as in example 2.17. ⌟

2.6. **Identity types.** We now recall the definitions of some classes of identity types. We only use identity types with an elimination rule *à la Paulin-Mohring*, also called based path induction. We only work with *weak* identity types, whose computation rule only holds up to a path.

We use both strictly stable and weakly stable variants of the identity type. In presence of either variant, we have well-behaved notions of contractibility, equivalence, transport, etc. that we don't expliciely introduce.

2.6.1. *Weak identity types.*

Definition 2.23 (Weak identity types)**.** The structure of **weak identity types** over an internal family $(\mathsf{Ty}, \mathsf{Tm})$ consists of four components Id, refl, J, $\mathsf{J}\beta$ with the following signature:

$$\begin{aligned}
&\mathsf{Id} \;\;: \forall (A : \mathsf{Ty})\ (x, y : \mathsf{Tm}(A)) \to \mathsf{Ty},\\
&\mathsf{refl} : \forall (A : \mathsf{Ty})\ (x : \mathsf{Tm}(A)) \to \mathsf{Tm}(\mathsf{Id}(A, x, x)),\\
&\mathsf{J} \;\;\; : \forall (A : \mathsf{Ty})\ (x : \mathsf{Tm}(A))\\
&\qquad (P : (y : \mathsf{Tm}(A))(p : \mathsf{Tm}(\mathsf{Id}(A, x, y))) \to \mathsf{Ty})\ (d : \mathsf{Tm}(P(x, \mathsf{refl}(x))))\\
&\qquad \to \forall y\ p \to \mathsf{Tm}(P(y, p)),\\
&\mathsf{J}\beta \;: \forall A\ x\ P\ d \to \mathsf{Tm}(\mathsf{Id}(P(x, \mathsf{refl}(x)), \mathsf{J}(A, x, P, d, x, \mathsf{refl}(x)), d)).
\end{aligned}$$

⌟

Once a CwF has weak identity types, many notions can be derived, such as composition of paths, the action on paths of a function, the notion of contractibility, etc. They can be defined mostly in the same way as in the HoTT book (Univalent Foundations Program 2013), although some additional effort is needed to deal with the absence of the strict β-rule for J and with the lack of Σ- and Π- types.

2.6.2. *Weakly stable identity types.* We will only consider weakly stable identity types with a weak computation rule. We fix a base CwF $\mathcal{C}$.

Definition 2.24 (Weakly stable identity types)**.** A Id**-introduction context** is a triple (Γ, A, x), where

$$\begin{aligned}
&\Gamma \;:\; \mathsf{Ob}_{\mathcal{C}},\\
&A :: \mathbf{y}(\Gamma) \to \mathsf{Ty},\\
&x \;:: (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}(A(\gamma)).
\end{aligned}$$

Here Γ is an object of $\mathcal{C}$, and A and x are types and terms that only depend on Γ.

A **weakly stable identity type introduction structure** consists, for every Id-introduction context (Γ, A, x), of operations

$$\begin{aligned}
&\mathsf{Id}_{(\Gamma,A,x)} \;\; :: \forall (\gamma : \mathbf{y}(\Gamma))(y : \mathsf{Tm}(A(\gamma))) \to \mathsf{Ty},\\
&\mathsf{refl}_{(\Gamma,A,x)} :: \forall (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}(\mathsf{Id}_{(\Gamma,A,x)}(\gamma, x)).
\end{aligned}$$

A Id**-elimination context** over an Id-introduction context (Γ, A, x) is a tuple (Δ, γ, P, d), where

$$\begin{aligned}
&\Delta :\; \mathsf{Ob}_{\mathcal{C}},\\
&\gamma :\; \Delta \to \Gamma,\\
&P :: \forall (\delta : \mathbf{y}(\Delta))(y : \mathsf{Tm}(A(\gamma(\delta))))(p : \mathsf{Tm}(\mathsf{Id}_{(\Gamma,A,x)}(\gamma(\delta), y))) \to \mathsf{Ty},\\
&d \;:: \forall (\delta : \mathbf{y}(\Delta)) \to \mathsf{Tm}(P(\delta, x(\gamma(\delta)), \mathsf{refl}_{(\Gamma,A,x)}(\gamma(\delta), x(\gamma(\delta))))).
\end{aligned}$$

A **weakly stable identity type elimination structure** consists, for every Id-elimination context (Δ, γ, P, d) over (Γ, A, x), of operations

$$\begin{aligned}
&\mathsf{J}_{(\Gamma,A,x,\Delta,\gamma,P,d)} \;:: \;\forall (\delta : \mathbf{y}(\Delta))(y : \mathsf{Tm}(A(\gamma(\delta))))(p : \mathsf{Tm}(\mathsf{Id}_{(\Gamma,A,x)}(\gamma(\delta), y))) \to \mathsf{Tm}(P(\delta, y, p)),\\
&\mathsf{J}\beta_{(\Gamma,A,x,\Delta,\gamma,P,d)} :: \;\forall (\delta : \mathbf{y}(\Delta)) \to \mathsf{Tm}(\mathsf{Id}_{(\Delta,P',d)}(\delta, \mathsf{J}_{(\Gamma,A,x,\Delta,\gamma,P,d)}(\delta, x(\gamma(\delta)), \mathsf{refl}_{(\Gamma,A,x)}(\gamma(\delta))))),\\
&P'(\delta') \qquad\qquad \triangleq P(\delta', x(\delta'), \mathsf{refl}_{\Gamma}(\gamma(\delta'), x(\delta'))).
\end{aligned}$$

A **weakly stable identity type structure** consists of introduction and elimination structures. ⌟

Definition 2.25. Let $\mathcal{C}$ be a CwF that is equipped with weakly stable identity types. Given a type $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$, the set $\mathrm{isContr}(A)$ of witnesses of contractibility of A is defined as

$$\mathrm{isContr}(A) \triangleq (\forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(A)) \times (\forall \gamma\ (x, y : \mathsf{Tm}_{\mathcal{C}}(A)) \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_{(\Gamma.A,A)}((\gamma, x), y))).$$

⌟

Definition 2.26. Let $\mathcal{C}$ be a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF that is also equipped with weakly stable identity types. We say that $\mathcal{C}$ satisfies **function extensionality** if for every $\Gamma : \mathsf{Ob}_{\mathcal{C}}$, $A :: \mathbf{y}(\Gamma) \to \mathsf{RepTy}_{\mathcal{C}}$, $B :: (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma)) \to \mathsf{Ty}_{\mathcal{C}}$ and $f :: (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}((a : A(\gamma)) \to B(\gamma, a))$, the type

$$(g : (a : A(\gamma)) \to B(\gamma, a)) \times ((a : A(\gamma)) \to \mathsf{Id}_{((\gamma':\Gamma).(a':A(\gamma')),B(\gamma',a'),f(\gamma',a'))}((\gamma, a), g(a)))$$

is contractible (over $(\gamma : \Gamma)$). ⌟

We write $\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}},\mathsf{Id}_{ws}}$ for the category of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs equipped with weakly stable identity types that satisfy function extensionality. A $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwF can be thought of as an ∞-category with representable maps.

3. Second-order generalized algebraic theories

We introduce our definition of second-order generalized algebraic theory (SOGAT), which are algebraic theories with dependent sorts and bindings. It is closely related to Uemura's general definition of type theory with functorial semantics in representable map categories (Uemura 2019); a large part of the material presented in this section can be found in Uemura's work, with a different presentation. We call these theories SOGATs rather than type theories to emphasize that we also consider theories that are not usually seen as type theories, such as the (first-order) generalized algebraic theory of categories. We note that Uemura uses the term SOGAT to refer to syntactic presentations of representable map categories in his thesis (Uemura 2021).

Our definition differs from Uemura's definition in the following ways:

- Uemura's representable map categories have all finite limits. This means that they generalize essentially algebraic theories (EATs) rather than generalized algebraic theories (GATs). Essentially algebraic theories do not have dependent sorts, but allow for partial operations instead. Any generalized algebraic theory induces an essentially algebraic theory with an equivalent category of models, but this translation loses information about the sort dependencies. This information is important; for example it equips the category of models of a generalized algebraic theory with notions of cofibrations and trivial fibrations (see section 3.4).
- We use $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs instead of representable map categories. This is partially a matter of preference, as $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs ought to be equivalent to categories with classes of representable maps and display maps ("representable map clans"). One advantage of our approach is that freely generated $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs are perhaps easier to understand syntactically, since they are themselves the initial models of some type theories. Furthermore, we may embed $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs into CwFs with additional structure. In particular we will consider $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwFs, which should correspond to some notion of representable map ∞-categories. It seems possible to observe both homotopical and computational properties of the theories using $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwFs, while computational properties are not always easily observable with ∞-categories (depending on the chosen model of ∞-categories).
- We prefer to work with the 1-category of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs and strict $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphisms, instead of the $(2, 1)$-category of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs and pseudo-morphisms. Similarly, we prefer to work with its 1-category of models and strict morphisms, rather than the $(2, 1)$-category of models and weak morphisms. One of the reason is that we consider factorization systems and semi model structures on these categories, which are easier to understand in the 1-categorical setting. This does not play an important role in this paper, as we work almost exclusively with the coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwF of the theory, without considering morphisms between other models.

3.1. Definition and functorial semantics.

Definition 3.1. A **second-order generalized algebraic theory** (SOGAT) is an $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}, E^{\mathsf{tm}}\}$-cellular $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$, where the maps $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}, E^{\mathsf{tm}}\}$ are the generic extensions of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs by a type, representable type, term or term equality:

$$\begin{aligned}
I^{\mathsf{ty}} &: \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}), \\
I^{\mathsf{ty}_{\mathsf{rep}}} &: \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}_{\mathsf{rep}}), \\
I^{\mathsf{tm}} &: \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{a} : \boldsymbol{A}), \\
E^{\mathsf{tm}} &: \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{x}, \boldsymbol{y} : \boldsymbol{A}) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{x} = \boldsymbol{y}).
\end{aligned}$$
⌟

In other words, a SOGAT is a presentation of a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF by collections of generating types, generating representable types, generating terms and generating equations between terms. We will write these generators using a **red, bold** font. In practice, a SOGAT is given by a signature, and the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ is reconstructed from the signature. We keep the notion of signature informal in this paper; a formal definition of signature can be given by modifying the definition of QIIT-signature of Kaposi, Kovács, and Altenkirch (2019). For every generating type, term or equation in a signature, the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF is extended by pushout against a map in $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}, E^{\mathsf{tm}}\}$.

For example, the signature of a pointed dependent type

$$\begin{aligned}
A &: \mathsf{Ty}, \\
B &: (a : \mathsf{Tm}(A)) \to \mathsf{Ty}, \\
b &: (a : \mathsf{Tm}(A)) \to \mathsf{Tm}(B(a)).
\end{aligned}$$

gets translated to the following iterated pushout:

$$\begin{tikzcd}
\mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash) \arrow[r, "\langle \diamond \rangle"'] \arrow[d] & \mathbf{0}_{\Sigma,\Pi_{\mathsf{rep}}} \arrow[d] & \\
\mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}) \arrow[r, "\langle \diamond, A \rangle"'] \arrow[ur, phantom, "\ulcorner"] & \mathbf{0}_{\Sigma,\Pi_{\mathsf{rep}}}[A] \arrow[d] & \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash) \arrow[l, "\langle (a{:}A) \rangle"] \arrow[d] \arrow[dl, phantom, "\urcorner"] \\
\mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}) \arrow[r, "\langle (a{:}A), B(a) \rangle"'] \arrow[d] & \mathbf{0}_{\Sigma,\Pi_{\mathsf{rep}}}[A, B] \arrow[d] & \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}) \arrow[l, "\langle (a{:}A), B(a) \rangle"] \\
\mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\boldsymbol{\Gamma} \vdash \boldsymbol{a} : \boldsymbol{A}\ \mathsf{type}) \arrow[r, "\langle (a{:}A), B(a), b(a) \rangle"'] & \mathbf{0}_{\Sigma,\Pi_{\mathsf{rep}}}[A, B, b]\ . &
\end{tikzcd}$$

Our running examples will be the first-order generalized algebraic theory $\mathcal{T}_{\mathbf{Cat}}$ of categories and the type theory $\mathcal{T}_{\mathsf{Id}}$ of weak identity types.

Example 3.2. The (first-order) generalized algebraic theory of categories $\mathcal{T}_{\mathbf{Cat}}$ is given by the following signature:

$$\begin{aligned}
&\mathbf{ob} && : \ \mathsf{Ty} \\
&\mathbf{Ob} && \triangleq \mathsf{Tm}(\mathbf{ob}) \\
&\mathbf{hom} && : \ \mathbf{Ob} \to \mathbf{Ob} \to \mathsf{Ty} \\
&\mathbf{Hom}(x,y) && \triangleq \mathsf{Tm}(\mathbf{hom}(x,y)) \\
&\mathbf{eqhom} && : \ \forall x\, y \to \mathbf{Hom}(x,y) \to \mathbf{Hom}(x,y) \to \mathsf{Ty} \\
&\mathbf{EqHom}(f,g) && \triangleq \mathsf{Tm}(\mathbf{eqhom}(f,g)) \\
&\mathbf{id} && : \ \forall x \to \mathbf{Hom}(x,x) \\
&\mathbf{comp} && : \ \forall x\, y\, z \to \mathbf{Hom}(x,y) \to \mathbf{Hom}(y,z) \to \mathbf{Hom}(x,z) \\
&f \circ g && \triangleq \mathbf{comp}(g,f) \\
&\mathbf{irefl} && : \ \forall x\, y \ \to (f : \mathbf{Hom}(x,y)) \to \mathbf{EqHom}(f,f) \\
& && \quad \mathbf{id} \circ f = f \\
& && \quad f \circ \mathbf{id} = f \\
& && \quad (f \circ g) \circ h = f \circ (g \circ h) \\
& && \quad \forall x\, y\, f\, g \to (p,q : \mathbf{EqHom}(f,g)) \to p = q \\
& && \quad \mathbf{EqHom}(f,g) \to f = g
\end{aligned}$$

We use the capitalized **Ob**, **Hom**, **EqHom** to denote the elements of the sorts **ob**, **hom** and **eqhom**. We also use $\_ \circ \_$ as an infix notation for composition.

Note that including the sort **eqhom** of equalities between morphisms does not change the categories of models of $\mathcal{T}_{\mathbf{Cat}}$. However it has to be included in order to determine the correct "language of categories". In our setting, including this sort is needed to equip $\mathcal{T}_{\mathbf{Cat}}$ with homotopy relations in section 4.1; isomorphisms cannot be defined without mentioning equality of morphisms. ⌟

Example 3.3. The SOGAT $\mathcal{T}_{\mathbf{CwF}}$ of a family with representable elements is given by the following signature:

$$\begin{aligned}
\mathbf{ity} \quad &: \mathsf{Ty} \\
\mathbf{iTy} \quad &\triangleq \mathsf{Tm}(\mathbf{ity}) \\
\mathbf{itm} \quad &: \mathbf{iTy} \to \mathsf{RepTy} \\
\mathbf{iTm}(A) &\triangleq \mathsf{Tm}(\mathbf{itm}(A))
\end{aligned}$$

Type-theoretic structures (Σ, Π, etc.) can be specified by extensions of this signature by new operations and equations. In particular, the theory $\mathcal{T}_{\mathsf{Id}}$ of weak identity types is the extension of $\mathcal{T}_{\mathbf{CwF}}$ by the new operations $(\mathbf{iId}, \mathbf{irefl}, \mathbf{iJ}, \mathbf{iJ}_\beta)$ with the signature given in definition 2.23. ⌟

For the remainder of this section, we fix an arbitrary SOGAT $\mathcal{T}$.

We now briefly recall the main definitions of the functorial semantics of $\mathcal{T}$; we refer the reader to Uemura 2019 for further details. The main results of this paper only involve the syntax of $\mathcal{T}$; but are motivated by the semantics.

Definition 3.4. An **internal model** of $\mathcal{T}$ in a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphism

$$\mathbb{C} : \mathcal{T} \to \mathcal{C}.$$ ⌟

When unambiguous, we will write $X_{\mathcal{C}}$, $A_{\mathcal{C}}$, $a_{\mathcal{C}}$, etc. instead of $\mathbb{C}(X)$, $\mathbb{C}(A)$, $\mathbb{C}(a)$, etc. for the application of the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphism $\mathbb{C}$ on objects, morphisms, types and terms.

By the universal property of $\mathcal{T}$, an internal model in $\mathcal{C}$ is uniquely determined by the image of the generators of $\mathcal{T}$, that is by an interpretation of the signature $\mathcal{T}$ in $\mathcal{C}$.

We have a locally finitely presentable 1-category $(\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}}} \backslash \mathcal{T})$ of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs equipped with an internal model of $\mathcal{T}$. The identity morphism $\mathsf{id} : \mathcal{T} \to \mathcal{T}$ equips $\mathcal{T}$ with the structure of an internal model, called the *generic model* of $\mathcal{T}$. It is also the initial object of $(\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}}} \backslash \mathcal{T})$).

Definition 3.5. A **model** of $\mathcal{T}$ consists of a category $\mathcal{C}$ with a terminal object, along with an internal model of $\mathcal{T}$ in the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\widehat{\mathcal{C}}$, that is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphism $\mathbb{C} : \mathcal{T} \to \widehat{\mathcal{C}}$. ⌟

Definition 3.6. A **weak morphism** F of models of $\mathcal{T}$ consists of a functor $F : \mathcal{C} \to \mathcal{D}$ such that:

- The functor F weakly preserves terminal objects.
- For every object $X : \mathcal{T}$, we have a transformation
$$F^X : (\Gamma : \mathcal{C}^{\mathsf{op}}) \to (\mathsf{y}(\Gamma) \to X_{\mathcal{C}}) \to (\mathsf{y}(F(\Gamma)) \to X_{\mathcal{D}}),$$
contravariantly natural in Γ.
- For every morphism $\alpha : X \to Y$, the following square commutes
$$\begin{tikzcd}
(\mathsf{y}(\Gamma) \to X_{\mathcal{C}}) \arrow[r, "F^X"] \arrow[d, "(\alpha_{\mathcal{C}} \circ -)"] & (\mathsf{y}(F(\Gamma)) \to X_{\mathcal{D}}) \arrow[d, "(\alpha_{\mathcal{D}} \circ -)"] \\
(\mathsf{y}(\Gamma) \to Y_{\mathcal{C}}) \arrow[r, "F^Y"] & (\mathsf{y}(F(\Gamma)) \to Y_{\mathcal{D}})
\end{tikzcd}$$
- Remark that we obtain, for every object $X : \mathcal{T}$ and type $A :: \mathsf{y}(X) \to \mathsf{Ty}_{\mathcal{T}}$, a natural transformation
$$\begin{aligned}
F^A : (\Gamma : \mathcal{C}^{\mathsf{op}}) \to (x : \mathsf{y}(\Gamma) \to X_{\mathcal{C}}) \to (a : (\gamma : \mathsf{y}(\Gamma)) \to A_{\mathcal{C}}(x(\gamma))) \\
\to ((\gamma : \mathsf{y}(F(\Gamma))) \to A_{\mathcal{D}}(F^X(x)(\gamma)))
\end{aligned}$$
such that $F^{X.A}(x, a) = (F^X(x), F^A(x, a))$.

- Context extensions are weakly preserved: for every object $X : \mathcal{T}$, representable type $A :: \mathbf{y}(X) \to \mathsf{RepTy}_{\mathcal{T}}$, object $\Gamma : \mathcal{C}$ and element $x :: \mathbf{y}(\Gamma) \to X_{\mathcal{C}}$, the comparison map
$$\left\langle F(\boldsymbol{p}), F^{A}(\boldsymbol{q}) \right\rangle : F((\gamma : \Gamma).A_{\mathcal{C}}(x(\gamma))) \to (\gamma : F(\Gamma)).A_{\mathcal{D}}(F^{X}(x)(\gamma))$$
is an isomorphism.

A morphism is **strict** if the terminal object and context extensions are strictly preserved. ⌟

Definition 3.7. A 2-cell between two weak morphisms $F, G : \mathcal{C} \to \mathcal{D}$ of models of $\mathcal{T}$ consists of a natural isomorphism $\alpha : F \cong G$, such that:

- For every object $X : \mathcal{T}$, context $\Gamma : \mathcal{C}$, elements $x :: \mathbf{y}(\Gamma) \to X_{\mathcal{C}}$ and $\gamma : \mathbf{y}(F(\Gamma))$, we have $F^{X}(x, \gamma) = G^{X}(x, \alpha_{\Gamma}(\gamma))$. ⌟

We have a $(2,1)$-category of models, weak morphisms and 2-cells, and a 1-category $\mathbf{Mod}_{\mathcal{T}}$ of models and strict morphisms. We will mainly work with the 1-category $\mathbf{Mod}_{\mathcal{T}}$. The category $\mathbf{Mod}_{\mathcal{T}}$ is locally finitely presentable; in particular we have an initial model $\mathbf{0}_{\mathcal{T}}$ and more general freely generated models.

3.2. Structure of the types of a SOGAT.

We write $\mathsf{GenTy}_{\mathcal{T}}$ for the set of generating types of $\mathcal{T}$; it can be obtained from the presentation of $\mathcal{T}$ as an $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}, E^{\mathsf{tm}}\}$-cellular $(\Sigma, \Pi_{\mathsf{rep}})$-CwF. For every $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, we have an object $\partial\mathbf{S} : \mathcal{T}$ and a type $\mathbf{S} : \mathbf{y}(\partial\mathbf{S}) \to \mathsf{Ty}_{\mathcal{T}}$.

We also have a subset $\mathsf{GenRepTy}_{\mathcal{T}} \subseteq \mathsf{GenTy}_{\mathcal{T}}$ of generating representable types of $\mathcal{T}$.

For example, $\mathsf{GenTy}_{\mathcal{T}_{\mathsf{Cat}}} = \{\mathbf{ob}, \mathbf{hom}\}$ with $\partial\mathbf{ob} = \mathbf{1}$ and $\partial\mathbf{hom} = \mathbf{ob} \times \mathbf{ob}$. For $\mathcal{T}_{\mathsf{Id}}$, we have $\mathsf{GenTy}_{\mathcal{T}_{\mathsf{Id}}} = \{\mathbf{ity}, \mathbf{itm}\}$ and $\mathsf{GenRepTy}_{\mathcal{T}_{\mathsf{Id}}} = \{\mathbf{itm}\}$, with $\partial\mathbf{ity} = \mathbf{1}$ and $\partial\mathbf{itm} = \mathbf{ity}$.

Because a SOGAT cannot contain any equations between sorts, the types of $\mathcal{T}$ can all be reconstructed by closing the generating types under substitution and the type-formers Σ, $\mathbf{1}$ and Π. We can consider the same closure in arbitrary internal models of $\mathcal{T}$. Furthermore we stratify these types into basic types (obtained by closing the generating types under substitution), the monomial types (obtained by closing the basic types under dependent products with arities in representable types) and the polynomial types ("sums of products", obtained by closing the monomial types under dependent sums).

Definition 3.8. Let $\mathcal{C}$ be a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF equipped with an internal model of $\mathcal{T}$. We define families $\mathsf{BTy}_{\mathcal{C}}$, $\mathsf{MonoTy}_{\mathcal{C}}$ and $\mathsf{PolyTy}_{\mathcal{C}}$ that are restrictions of $\mathsf{Ty}_{\mathcal{C}}$, and a restricted family $\mathsf{BRepTy}_{\mathcal{C}} \to \mathsf{RepTy}_{\mathcal{C}}$. We work internally to $\mathbf{Psh}(\mathcal{C})$.

- A **basic type** $\mathbf{S}(\sigma) : \mathsf{BTy}_{\mathcal{C}}$ consists of $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$ and $\sigma : \mathsf{Tm}_{\mathcal{C}}(\partial\mathbf{S}_{\mathcal{C}})$.
  The corresponding type in $\mathsf{Ty}_{\mathcal{C}}$ is $\mathbf{S}_{\mathcal{C}}(\sigma)$.
- A **basic representable type** $\mathbf{S}(\sigma) : \mathsf{BRepTy}_{\mathcal{C}}$ consists of $\mathbf{S} : \mathsf{GenRepTy}_{\mathcal{T}}$ and $\sigma : \mathsf{Tm}_{\mathcal{C}}(\partial\mathbf{S})$.
- A **monomial type** $[\Delta \vdash A] : \mathsf{MonoTy}_{\mathcal{C}}$ consists of a telescope $\Delta : \mathsf{BRepTy}^{\star}_{\mathcal{C}}$ of basic representable types, along with a dependent basic type $A : \mathsf{Tm}^{\star}_{\mathcal{C}}(\Delta) \to \mathsf{BTy}_{\mathcal{C}}$. The corresponding type in $\mathsf{Ty}_{\mathcal{C}}$ is an iterated first-order Π-type.
- A **polynomial type** is a telescope of monomial types: $\mathsf{PolyTy}_{\mathcal{C}} \triangleq \mathsf{MonoTy}^{\star}_{\mathcal{C}}$. The corresponding type in $\mathsf{Ty}_{\mathcal{C}}$ is obtained as an iterated Σ-type. ⌟

We also define the closure $\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$ of basic types under $\mathbf{1}$-, Σ- and first-order Π- types.

Definition 3.9. We define restricted families $\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}}) \to \mathsf{Ty}_{\mathcal{C}}$ and $\mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}}) \to \mathsf{RepTy}_{\mathcal{C}}$ by induction-recursion (internally to $\mathbf{Psh}(\mathcal{C})$).
$$\begin{aligned}
&\mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}}) : \mathcal{U}, \\
&\iota_{\mathsf{rep}} \quad : \mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}}) \to \mathsf{RepTy}_{\mathcal{C}}, \\
&\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}}) : \mathcal{U}, \\
&\iota \quad : \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}}) \to \mathsf{Ty}_{\mathcal{C}}.
\end{aligned}$$
The family $\mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}})$ has constructors $\tau_{\mathsf{rep}} : \mathsf{BRepTy}_{\mathcal{C}} \to \mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}})$, $\mathbf{1}$ and Σ, with $\iota_{\mathsf{rep}}(\tau_{\mathsf{rep}}(A)) = A$, $\iota_{\mathsf{rep}}(\mathbf{1}) = \mathbf{1}$ and $\iota_{\mathsf{rep}}(\Sigma(A, B)) = \Sigma(\iota_{\mathsf{rep}}(A), \lambda a \mapsto \iota_{\mathsf{rep}}(B(a)))$. Similarly, the family $\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$ has constructors $\tau : \mathsf{BTy}_{\mathcal{C}} \to \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$, $- : \mathsf{RepTy}_{\mathcal{C}} \to \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$, $\mathbf{1}$, Σ and Π_{rep} that are preserved by ι. ⌟

Proposition 3.10. *The canonical maps* $\mathsf{PolyTy}_{\mathcal{C}} \to \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$ *and* $\mathsf{BRepTy}^{\star}_{\mathcal{C}} \to \mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}})$ *are essentially surjective: for every* $A : \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{C}})$*, there is some* $A_0 : \mathsf{PolyTy}_{\mathcal{C}}$ *such that* $\mathsf{Tm}_{\mathcal{C}}(A_0) \simeq \mathsf{Tm}_{\mathcal{C}}(A)$*; and for every* $A : \mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{C}})$*, there is some* $A_0 : \mathsf{BRepTy}^{\star}_{\mathcal{C}}$ *such that* $\mathsf{Tm}_{\mathcal{C}}(A_0) \simeq \mathsf{Tm}_{\mathcal{C}}(A)$*.*

Proof. This follows from the facts that Σ-types are essentially associative and that (first-order) Π-types essentially distribute over Σ-types. □

Since the presentation of $\mathcal{T}$ does not include any type equation, the types of $\mathcal{T}$ are exactly the closure of the basic types under $\mathbf{1}$, Σ and first-order Π -types.

Proposition 3.11. *The canonical maps*

$$\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{T}}) \to \mathsf{Ty}_{\mathcal{T}}$$

and

$$\mathsf{Clos}_{\Sigma}(\mathsf{BRepTy}_{\mathcal{T}}) \to \mathsf{RepTy}_{\mathcal{T}}$$

are isomorphisms. □

We omit the proof; it follows from a standard normalization argument. This result allows us to use induction over the structure of types of $\mathcal{T}$.

Corollary 3.12. *The canonical maps* $\mathsf{PolyTy}_{\mathcal{T}} \to \mathsf{Ty}_{\mathcal{T}}$ *and* $\mathsf{BRepTy}^{\star}_{\mathcal{T}} \to \mathsf{RepTy}_{\mathcal{T}}$ *are essentially surjective: for every* $A : \mathsf{Ty}_{\mathcal{T}}$*, there is some* $A_0 : \mathsf{PolyTy}_{\mathcal{T}}$ *such that* $A_0 \simeq A$*; and for every* $A : \mathsf{RepTy}_{\mathcal{T}}$*, there is some* $A_0 : \mathsf{BRepTy}^{\star}_{\mathcal{T}}$ *such that* $A_0 \simeq A$*.*

Proof. By proposition 3.10 and proposition 3.11. □

Proposition 3.13. *The family restriction* $\mathsf{RepTy}_{\mathcal{T}} \to \mathsf{Ty}_{\mathcal{T}}$ *is a monomorphism.*

Proof. This follows from the isomorphism $\mathsf{Ty}_{\mathcal{T}} \cong \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{T}})$. Indeed $\mathsf{RepTy}_{\mathcal{T}} \to \mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{T}})$ is a constructor of $\mathsf{Clos}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathsf{BTy}_{\mathcal{T}})$, and is therefore injective. □

3.3. Contextual models.

We can generalize the notions of contextuality from CwFs to the category of models of an arbitrary SOGAT.

Definition 3.14. A morphism $F : \mathcal{C} \to \mathcal{D}$ in $\mathbf{Mod}_{\mathcal{T}}$ is a **contextual isomorphism** if it is bijective on every sort: for every generating type $\mathsf{S} : \mathsf{GenTy}_{\mathcal{T}}$, object $\Gamma : \mathcal{C}$, boundary $\sigma :: \mathsf{y}(\Gamma) \to \partial\mathsf{S}_{\mathcal{C}}$ and element $a :: (\gamma : \mathsf{y}(F(\Gamma))) \to \mathsf{S}_{\mathcal{D}}(F(\sigma)(\gamma))$, there is a unique element $a_0 :: (\gamma : \mathsf{y}(\Gamma)) \to \mathsf{S}_{\mathcal{C}}(\sigma(\gamma))$ such that $F(a_0) = a$. ⌟

The contextual isomorphisms are the right class of maps of an orthogonal factorization system generated by a set $I_{\mathcal{T}}$ of maps in $\mathbf{Mod}_{\mathcal{T}}$.

$$\begin{aligned} I_{\mathcal{T}} &\triangleq \{ I^{\mathsf{S}} \mid \mathsf{S} : \mathsf{GenTy}_{\mathcal{T}} \} \\ I^{\mathsf{S}} &: \ \mathsf{Free}_{\mathcal{T}}(\boldsymbol{\Gamma} \vdash \boldsymbol{\sigma} : \partial\mathsf{S}) \to \mathsf{Free}_{\mathcal{T}}(\boldsymbol{\Gamma} \vdash \boldsymbol{x} : \mathsf{S}(\boldsymbol{\sigma})) \end{aligned}$$

The maps in the corresponding left class are called **left contextual** maps.

Definition 3.15. The **contextual core** $\mathsf{cxl}(\mathcal{C})$ is obtained from the factorization of the unique map $\mathbf{0}_{\mathcal{T}} \to \mathcal{C}$ as a left contextual map $\mathbf{0}_{\mathcal{T}} \to \mathsf{cxl}(\mathcal{C})$ followed by a contextual isomorphism $\mathsf{cxl}(\mathcal{C}) \to \mathcal{C}$. ⌟

Definition 3.16. A model $\mathcal{C} : \mathbf{Mod}_{\mathcal{T}}$ is **contextual** if $\mathsf{cxl}(\mathcal{C}) \to \mathcal{C}$ is an isomorphism. ⌟

The 1-category $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}}$ of contextual models forms a coreflective subcategory of $\mathbf{Mod}_{\mathcal{T}}$; the functor $\mathsf{cxl} : \mathbf{Mod}_{\mathcal{T}} \to \mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}}$ is right adjoint to the subcategory inclusion $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}} \to \mathbf{Mod}_{\mathcal{T}}$.

3.4. **Trivial fibrations.**

Definition 3.17. A morphism $F : \mathcal{C} \to \mathcal{D}$ in $\mathbf{Mod}_{\mathcal{T}}$ is a **trivial fibration** if it is surjective on every sort: for every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, object $\Gamma : \mathcal{C}$, boundary $\sigma : よ(\Gamma) \to \partial\mathbf{S}_{\mathcal{C}}$ and element $a : (\gamma : よ(F(\Gamma))) \to \mathbf{S}_{\mathcal{D}}(F(\sigma)(\gamma))$, there exists an element $a_0 : (\gamma : よ(\Gamma)) \to \mathbf{S}_{\mathcal{C}}(\sigma(\gamma))$ such that $F(a_0) = a$. ⌟

The trivial fibrations are the right class of maps of the weak factorization system that is cofibrantly generated by the same set $I_{\mathcal{T}}$ of maps that we used to define contextual isomorphisms. The maps in the left class are called **cofibrations**.

In the case of the GAT $\mathcal{T}_{\mathbf{Cat}}$, the trivial fibrations are the trivial fibrations of the canonical model structure on **Cat**, that is functors that are surjective on objects and fully faithful.

For the SOGAT $\mathcal{T}_{\mathbf{CwF}}$, the (cofibrations, trivial fibrations) weak factorization system on **CwF** coincides with the one defined by Kapulkin and Lumsdaine (2018).

4. Theories with homotopy relations

4.1. **Homotopy relations.**

We now consider SOGATs that are equipped with an additional piece of data: a choice of a homotopy relation for every generating sort of the theory.

From the point of view of model categories, this roughly corresponds to the choice of a relative cylinder object for every generating cofibration.

Definition 4.1. The data of **homotopy relations** on a SOGAT $\mathcal{T}$ consists, for every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, of a reflexive type-valued binary relation on its terms:

$$\begin{aligned} \_ \sim_{\mathbf{S}(\_)} \_ &:: (\sigma : \partial\mathbf{S})(x, y : \mathbf{S}(\sigma)) \to \mathsf{Ty}_{\mathcal{T}}, \\ \mathsf{refl}_{\mathbf{S}(\_)} &:: (\sigma : \partial\mathbf{S})(x : \mathbf{S}(\sigma)) \to x \sim_{\mathbf{S}(\sigma)} x. \end{aligned}$$ ⌟

Since these homotopy relations are specified in the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$, they are automatically available in any other model of $\mathcal{T}$.

Example 4.2. Homotopy relations are defined over the theory of categories $\mathcal{T}_{\mathbf{Cat}}$ as follows:

$$\begin{aligned} x \sim_{\mathbf{ob}} y &\triangleq (x \cong y), \\ f \sim_{\mathbf{hom}(x,y)} g &\triangleq \mathbf{eqhom}(f, g), \\ \_ \sim_{\mathbf{eqhom}(f,g)} \_ &\triangleq \mathbf{1}, \end{aligned}$$

where $(x \cong y)$ is the type of isomorphisms between x and y, i.e.

$$\begin{aligned} (x \cong y) \triangleq (f : \mathbf{hom}(x, y)) \times (g : \mathbf{hom}(y, x)) \\ \times\, \mathbf{eqhom}(g \circ f, \mathbf{id}) \times \mathbf{eqhom}(f \circ g, \mathbf{id}). \end{aligned}$$

Reflexivities are given by the identity isomorphisms on objects, by $\mathbf{irefl}$ on morphisms, and by tt on equalities between morphisms. ⌟

Example 4.3. Homotopy relations are defined over the type theory $\mathcal{T}_{\mathsf{Id}}$ of weak identity types as follows:

$$\begin{aligned} A \sim_{\mathbf{ity}} B &\triangleq \mathrm{iEquiv}(A, B) \\ x \sim_{\mathbf{itm}(A)} y &\triangleq \mathbf{itm}(\mathbf{iId}(A, x, y)) \end{aligned}$$

where $\mathrm{iEquiv}(A, B)$ is the type of relational equivalences between A and B. Note that even though $\mathrm{iEquiv}(A, B)$ is not classified by an inner type in $\mathcal{T}_{\mathsf{Id}}$, it can be written as an outer type in $\mathcal{T}_{\mathsf{Id}}$.

$$\begin{aligned} \mathrm{Equiv}(A, B) &\triangleq (R : \mathbf{itm}(A) \to \mathbf{itm}(B) \to \mathbf{ity}) \\ &\quad \times ((a : \mathbf{itm}(A)) \to \mathrm{isContr}((b : B) \times R(a, b))) \\ &\quad \times ((b : \mathbf{itm}(B)) \to \mathrm{isContr}((a : A) \times R(a, b))) \\ \mathrm{isContr}(X) &\triangleq (x : \mathbf{itm}(X)) \times (\forall (x, y : \mathbf{itm}(X)) \to \mathbf{itm}(\mathbf{iId}(X, x, y))) \end{aligned}$$

Reflexivities are given by the identity equivalence $\mathbf{iId}_{\_}$ on types, and by $\mathbf{irefl}$ on terms. ⌟

We fix a SOGAT $\mathcal{T}$ equipped with homotopy relations for the remainder of this section.

4.2. Classes of maps.

The homotopy relations induce notions of weak equivalences and of fibrations over the category $\mathbf{Mod}_{\mathcal{T}}$. In the case of the theory $\mathcal{T}_{\mathsf{Id}}$ of weak identity types, we recover the classes of weak equivalences and fibrations on $\mathbf{CwF}_{\mathsf{Id}}$ that were introduced by Kapulkin and Lumsdaine (2018).

Definition 4.4. A morphism $F : \mathcal{C} \to \mathcal{D}$ in $\mathbf{Mod}_{\mathcal{T}}$ is a **weak equivalence** if it is essentially surjective on every sort: for every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, object $\Gamma : \mathcal{C}$, boundary $\sigma :: \mathrm{y}(\Gamma) \to \partial\mathbf{S}_{\mathcal{C}}$, and element $x :: (\gamma : \mathrm{y}(F(\Gamma))) \to \mathbf{S}_{\mathcal{D}}(F(\sigma)(\gamma))$, there exists a lifted element $x_0 :: (\gamma : \mathrm{y}(\Gamma)) \to \mathbf{S}_{\mathcal{C}}(\sigma(\gamma))$ along with a homotopy

$$p :: (\gamma : \mathrm{y}(F(\Gamma))) \to F(x_0)(\gamma) \sim_{\mathbf{S}_{\mathcal{D}}(F(\sigma)(\gamma))} x(\gamma).$$

⌟

Definition 4.5. A morphism $F : \mathcal{C} \to \mathcal{D}$ in $\mathbf{Mod}_{\mathcal{T}}$ is a **fibration** if it satisfies a lifting condition for homotopies with a fixed left endpoint.

homotopy lifting: For every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, object $\Gamma : \mathcal{C}$, boundary $\sigma :: \mathrm{y}(\Gamma) \to \partial\mathbf{S}_{\mathcal{C}}$, element $x :: (\gamma : \mathrm{y}(\Gamma)) \to \mathbf{S}_{\mathcal{C}}(\sigma(\gamma))$ and homotopy

$$p :: (\gamma : \mathrm{y}(F(\Gamma))) \to F(x)(\gamma) \sim_{\mathbf{S}_{\mathcal{D}}(F(\sigma)(\gamma))} y(\gamma),$$

there exists a homotopy

$$p_0 :: (\gamma : \mathrm{y}(\Gamma)) \to x(\gamma) \sim_{\mathbf{S}_{\mathcal{C}}(\sigma(\gamma))} y_0(\gamma)$$

such that $F(y_0) = y$ and $F(p_0) = p$. ⌟

4.3. Univalent internal models.

Recall that a $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwF is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF equipped with weakly stable identity types satisfying function extensionality. Consider a $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwF $\mathcal{C}$ equipped with an internal model of $\mathcal{T}$. Internally to $\mathcal{C}$, we have two notions of "weak equality" between elements of the model of $\mathcal{T}$, given by the homotopy relations $(\sim)$ and by the (outer) identity types $(\simeq)$. There is always a comparison map that sends elements of the outer identity types $(\simeq)$ to homotopies $(\sim)$, defined by sending the outer reflexivity to the inner reflexivity. It is then natural to ask for this map to be an equivalence (with respect to the outer identity types). We express this as a contractibility condition.

Definition 4.6. Let $\mathcal{C}$ be a $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwF equipped with an internal model of $\mathcal{T}$. We say that the internal model is **univalent**, or that the identity types are **saturated** (with respect to the homotopy relations) if for every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, the dependent type

$$(y : \mathbf{S}_{\mathcal{C}}(\sigma)) \times (p : x \sim_{\mathbf{S}_{\mathcal{C}}(\sigma)} y)$$

is contractible over $(\sigma : \partial\mathbf{S}_{\mathcal{C}}, x : \mathbf{S}_{\mathcal{C}}(\sigma))$, for the notion of contractibility induced by the outer identity types $(\simeq)$. ⌟

In the case of the theory $\mathcal{T}_{\mathbf{Cat}}$, an internal category is univalent in the sense of definition 4.6 when it is univalent in the sense of HoTT (Ahrens, Kapulkin, and Shulman 2015).

4.4. External univalence.

We can finally define the main notion of this paper.

Definition 4.7. We say that a SOGAT $\mathcal{T}$ equipped with homotopy relations satisfies **external univalence** when the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ can be equipped with weakly stable identity types satisfying function extensionality and saturation with respect to the homotopy relations. ⌟

The following claim will be proven in a future paper.

Claim 4.8. Let $\mathcal{T}$ be a SOGAT equipped with homotopy relations. It satisfies external univalence if and only if the category $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}}$ of contextual models of $\mathcal{T}$, equipped with the classes of trivial fibrations, fibrations and weak equivalences defined in definition 3.17, definition 4.5 and definition 4.4, is a left semi-model category.

5. Contractibility data and reflexive equivalences

We show that weakly stable identity types can be reconstructed from the data of reflexive relational equivalences (also called one-to-one relations or one-to-one correspondences). Similar ideas are used in the cubical type theory without an interval of Altenkirch and Kaposi (2015) and in the higher observational type theory of Altenkirch, Kaposi and Shulman (Shulman 2022; Altenkirch, Kaposi, and Shulman 2022).

We fix a CwF $\mathcal{C}$ equipped with Σ-types.

Definition 5.1 (Internally to $\mathbf{Psh}(\mathcal{C})$)**.** **Contractibility data** over $\mathcal{C}$ consists of a dependent presheaf

$$\mathrm{isContr} : \mathsf{Ty}_{\mathcal{C}} \to \mathcal{U}. \quad \lrcorner$$

Note that contractibility is not propositional data, even though a witness of contractibility should be unique up to homotopy. Whenever we say that some type A is contractible, we really mean that we have an element of $\mathrm{isContr}(A)$.

We now assume that $\mathcal{C}$ is equipped with global contractibility data.

Definition 5.2 (Internally to $\mathbf{Psh}(\mathcal{C})$)**.** An **equivalence** $E : A \simeq B$ between two types $A, B : \mathsf{Ty}_{\mathcal{C}}$ consists of a binary relation

$$E : \mathsf{Tm}_{\mathcal{C}}(A) \to \mathsf{Tm}_{\mathcal{C}}(B) \to \mathsf{Ty}_{\mathcal{C}}$$

that is functional in both directions, as witnessed by the following contractibility conditions

$$E.\overrightarrow{\mathsf{fun}} : \forall (a : \mathsf{Tm}_{\mathcal{C}}(A)) \to \mathrm{isContr}((b : B) \times E(a,b)),$$
$$E.\overleftarrow{\mathsf{fun}} : \forall (b : \mathsf{Tm}_{\mathcal{C}}(B)) \to \mathrm{isContr}((a : A) \times E(a,b)). \quad \lrcorner$$

Definition 5.3 (Internally to $\mathbf{Psh}(\mathcal{C})$)**.** A **reflexive equivalence** is an equivalence $E : A \simeq A$ that is additionally equipped with a reflexivity map

$$E.\mathsf{refl} : (a : \mathsf{Tm}_{\mathcal{C}}(A)) \to \mathsf{Tm}_{\mathcal{C}}(E(a,a)). \quad \lrcorner$$

Definition 5.4 (Internally to $\mathbf{Psh}(\mathcal{C})$)**.** A **dependent equivalence** for a dependent type $B : \mathsf{Tm}_{\mathcal{C}}(A) \to \mathsf{Ty}_{\mathcal{C}}$ over an equivalence $E : A \simeq A$ consists of a family of equivalences

$$T_B : \forall (x, y : \mathsf{Tm}_{\mathcal{C}}(A)) \to \mathsf{Tm}_{\mathcal{C}}(E(x,y)) \to B(x) \simeq B(y). \quad \lrcorner$$

Definition 5.5. We say that $\mathcal{C}$ is equipped with **reflexive equivalences** when for every type $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$, there is a reflexive equivalence

$$\mathsf{Id}_A :: \forall \gamma \to A(\gamma) \simeq A(\gamma).$$

We denote its reflexivity map by $\mathsf{refl}_A :: \forall \gamma\ (x : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Id}_A(\gamma, x, x)$. $\lrcorner$

Definition 5.6. We say that $\mathcal{C}$ is equipped with **dependent equivalences** when for every dependent type $B :: (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma)) \to \mathsf{Ty}_{\mathcal{C}}$, there is a dependent equivalence

$$\mathsf{DId}_{A.B} :: \forall \gamma\ (x, y : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma, x, y)) \to B(\gamma, x) \simeq B(\gamma, y). \quad \lrcorner$$

Note that we do not assume that $\mathsf{DId}_{A.B}$ is reflexive; the reason is that $\mathsf{DId}_{A.B}$ can be replaced by reflexive dependent equivalences by considering the composition

$$\mathsf{DId}_{A.B}(\gamma, x, y, p) \circ \mathsf{DId}_{A.B}(\gamma, x, x, \mathsf{refl}_A(x))^{-1}$$

when it is defined.

Definition 5.7. We say that a type $A :: \forall \gamma \to \mathsf{Ty}_{\mathcal{C}}$ has a **center** (of contraction) if we have an element

$$\mathsf{center}_A :: (\gamma : \mathbf{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma)).$$

Definition 5.8. We say that a type $A :: \forall \gamma \to \mathsf{Ty}_{\mathcal{C}}$ **has all paths**, or a **homogeneous all-paths operation**, if we have an element

$$\mathsf{hpath}_A :: \forall (\gamma : \mathbf{y}(\Gamma))\ (x, y : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma, x, y)).$$

In other words, the type A has all paths if it is a homotopy proposition, with respect to the identity type Id_A. $\lrcorner$

We will also need to consider an analogous heterogeneous structure for dependent contractible types, similarly to homogeneous and heterogeneous compositions structures in cubical type theories (see e.g. Angiuli et al. (2021)).

Definition 5.9. We say that a dependent type

$$B :: \forall(\gamma : \mathbf{y}(\Gamma))\ (a : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Ty}_{\mathcal{C}}$$

has a **heterogeneous all-paths operation** if we have

$$\begin{aligned}\mathsf{path}_B : \forall\gamma\ (a_l, a_r : \mathsf{Tm}_{\mathcal{C}}(A(\gamma)))\ (a_e : \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma, a_l, a_r)))\\ (b_l : \mathsf{Tm}_{\mathcal{C}}(B(\gamma, a_l)))\ (b_r : \mathsf{Tm}_{\mathcal{C}}(B(\gamma, a_r))) \to \mathsf{DId}_{A.B}(\gamma, a_e, b_l, b_r)).\end{aligned}$$ ⌟

We say that a contractibility witness

$$c :: \forall(\gamma : \mathbf{y}(\Gamma)) \to \mathrm{isContr}(A(\gamma))$$

has a center or a homogeneous all-paths operations if the type A has a center or a homogeneous all-paths operations. In that case, it is written $c.\mathsf{center}$ or $c.\mathsf{hpath}$.

Similarly, we say that a dependent contractibility witness

$$c :: \forall(\gamma : \mathbf{y}(\Gamma))\ (a : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathrm{isContr}(B(\gamma, a))$$

has a heterogeneous all-paths operations if the dependent type B has one. In that case, it is written $c.\mathsf{path}$.

When $c :: \forall\gamma \to \mathrm{isContr}((a : A(\gamma)) \times B(\gamma, a))$, the type $B(\gamma, a)$ is typically of the form $\mathsf{Id}_A(\gamma, a, a_0)$ or $\mathsf{Id}_A(\gamma, a_0, a)$ for some $a_0 : A(\gamma)$. In that case, we can think of the center and all-paths operations as specific cubical composition and filling operations, as described in the following diagrams (where we write $-.1$ and $-.2$ for the first and second projections out of a Σ-type):

$$\begin{tikzcd} a_0 \arrow[r, dash, dashed, "c.\mathsf{center}.2"] & c.\mathsf{center}.1 \end{tikzcd}$$

$$\begin{tikzcd} a_0 \arrow[r, equal] \arrow[d, dash, "b_l"'] \arrow[dr, phantom, "(c.\mathsf{hpath}.2)"] & a_0 \arrow[d, dash, "b_r"] \\ a_l \arrow[r, dash, dashed, "c.\mathsf{hpath}.1"'] & a_r \end{tikzcd}$$

Indeed, the first and second projections of the operations center and hpath correspond approximately to the operations coe, coh, uncoe and uncoh of the cubical type theory without an interval investigated by Altenkirch and Kaposi (2015).

Theorem 5.10. *Assume that $\mathcal{C}$ is equipped with the following data:*

- *A family* isContr *of contractibility data (definition 5.1);*
- *Along with reflexive equivalences $(\mathsf{Id}_-, \mathsf{refl}_-)$ (definition 5.5);*
- *Together with dependent equivalences (DId_-) (definition 5.6);*
- *Such that every contractible type has a center (definition 5.7) and all paths (definition 5.8).*

Then the identity type introduction structure $(\mathsf{Id}_-, \mathsf{refl}_-)$ can be equipped with a weakly stable elimination structure.

Proof. Let $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$ be a type of $\mathcal{C}$, along with a point $x :: \forall\gamma \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma))$.

Take parameters (Δ, γ, P, d) for the weakly stable elimination structure, consisting of:

$$\begin{aligned}&\Delta :\ \mathcal{C},\\ &\gamma :\ \Delta \to \Gamma,\\ &P :: \forall(\delta : \mathbf{y}(\Delta))(y : \mathsf{Tm}_{\mathcal{C}}(A(\gamma(\delta))))(p : \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma(\delta), x(\gamma(\delta)), y))) \to \mathsf{Ty}_{\mathcal{C}},\\ &d :: \forall(\delta : \mathbf{y}(\Delta)) \to \mathsf{Tm}_{\mathcal{C}}(P(x(\gamma(\delta)), \mathsf{refl}_A(\gamma(\delta), x(\gamma(\delta))))).\end{aligned}$$

We have to construct

$$\begin{aligned}
j &:: \forall(\delta : \mathbf{y}(\Delta))(y : \mathsf{Tm}_{\mathcal{C}}(A(\gamma(\delta))))(p : \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma(\delta), x(\gamma(\delta)), y))) \to \mathsf{Tm}_{\mathcal{C}}(P(\delta, y, p)),\\
j\beta &:: \forall(\delta : \mathbf{y}(\Delta)) \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_{\delta.P(\delta,x(\gamma(\delta)),\mathsf{refl}_A(\gamma(\delta),x(\gamma(\delta))))}(\delta, j(\delta, x(\gamma(\delta)), \mathsf{refl}_A(\gamma(\delta), x(\gamma(\delta)))), d(\delta))).
\end{aligned}$$

We pose $S(\delta) \triangleq (y : A(\gamma(\delta))) \times \mathsf{Id}_A(\gamma(\delta), x(\gamma(\delta)), y)$; it is the type of the dependency of the motive P. Since Id_A is a reflexive equivalence, S is a family of contractible types, i.e. we have an element $(\lambda\delta \mapsto \mathsf{Id}_A.\overrightarrow{\mathsf{fun}}(\gamma(\delta)))$ of $\forall\delta \to \mathsf{isContr}(S(\delta))$.

We now see P as a dependent type $P :: \forall\delta \to \mathsf{Tm}_{\mathcal{C}}(S(\delta)) \to \mathsf{Ty}_{\mathcal{C}}$. We consider the dependent equivalence

$$\mathsf{DId}_{S.P} :: \forall\delta\ (a, b : \mathsf{Tm}_{\mathcal{C}}(S(\delta)))\ (p : \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_S(\delta, a, b))) \to P(\delta, a) \simeq P(\delta, b).$$

Since S is contractible and contractible types have all paths, we can specialize $\mathsf{DId}_{S.P}$ to

$$T_{S.P} :: \forall\delta\ (a, b : \mathsf{Tm}_{\mathcal{C}}(S(\delta))) \to P(\delta, a) \simeq P(\delta, b),$$

providing a way to transport between different fibers of P.

We can now define j; we pose

$$\begin{aligned}
a &:: \forall\delta \to \mathsf{Tm}_{\mathcal{C}}(S(\delta)),\\
a(\delta) &\triangleq (x(\gamma(\delta)), \mathsf{refl}_A(\gamma(\delta), x(\gamma(\delta)))),\\
\overleftarrow{T_{S.P}} &: \forall\delta \to \mathsf{isContr}((d' : P(\delta, a(\delta))) \times T_{S.P}(\delta, a(\delta), a(\delta), d', d(\delta))),\\
\overleftarrow{T_{S.P}}(\delta) &\triangleq T_{S.P}(\delta, a(\delta), a(\delta)).\overleftarrow{\mathsf{fun}}(d(\delta)),\\
d' &: \forall\delta \to \mathsf{Tm}_{\mathcal{C}}(P(\delta, a(\delta))),\\
d' &\triangleq \overleftarrow{T_{S.P}}.\mathsf{center}(\delta).1,\\
\overrightarrow{T_{S.P}} &: \forall\delta\ (b : \mathsf{Tm}_{\mathcal{C}}(S(\delta))) \to \mathsf{isContr}((j : P(\delta, b(\delta))) \times T_{S.P}(\delta, a(\delta), b, d'(\delta), j)),\\
\overrightarrow{T_{S.P}}(\delta) &\triangleq T_{S.P}(\delta, a(\delta), b(\delta)).\overrightarrow{\mathsf{fun}}(d'(\delta)),\\
j &: \forall\delta \to (b : \mathsf{Tm}_{\mathcal{C}}(S(\delta))) \to \mathsf{Tm}_{\mathcal{C}}(P(\delta, b(\delta))),\\
j &\triangleq \overrightarrow{T_{S.P}}.\mathsf{center}(\delta, b).1.
\end{aligned}$$

Defining j by transporting twice deals with the lack of reflexivity for the dependent equivalences.

It remains to construct an element

$$j\beta :: \forall\delta \to \mathsf{Id}_{\delta.P(\delta,a(\delta))}(\delta, j(\delta, a(\delta)), d(\delta)).$$

Note that we have elements $\widetilde{d'} : T_{S.P}(\delta, a(\delta), a(\delta), d'(\delta), d(\delta))$ and $\widetilde{j} : T_{S.P}(\delta, a(\delta), a(\delta), d'(\delta), j(\delta, a(\delta)))$.

We consider the dependent type

$$\begin{aligned}
&Q(\delta, (y, q)) :\ \forall\delta\ (y : \mathsf{Tm}_{\mathcal{C}}(P(\delta, a(\delta))))\ (q : \mathsf{Tm}_{\mathcal{C}}(T_{S.P}(\delta, a(\delta), a(\delta), d'(\delta), y))) \to \mathsf{Ty}_{\mathcal{C}},\\
&Q(\delta, (y, q)) :: \mathsf{Id}_{\delta.P(\delta,a(\delta))}(\delta, y, d(\delta)).
\end{aligned}$$

The type of such pairs (y, q) is contractible, because $T_{S.P}$ is a dependent equivalence. Therefore, it has all paths and we obtain a family of equivalences

$$\forall\delta \to Q(\delta, (d(\delta), \widetilde{d'})) \simeq Q(\delta, (j(\delta, a(\delta)), \widetilde{j})).$$

The element $j\beta(\delta)$ is obtained by transporting $\mathsf{refl}_{\delta.P(\delta,a(\delta))}(\delta, d(\delta))$ over that equivalence. □

We also show that, conversely, any weakly stable identity type structures satisfies the assumptions of theorem 5.10. For this we need to construct contractibility data that is stable under substitution, that is we need to strictify the standard definition (definition 2.25) of contractible types.

Construction 5.11. Assume that $\mathcal{C}$ is equipped with weakly stable identity types. Then we construct a family $\mathsf{isContr}' : \mathsf{Ty}_{\mathcal{C}} \to \mathcal{U}$ such that for every $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$, there is a logical equivalence

$$(\forall\gamma \to \mathsf{isContr}'(A(\gamma))) \leftrightarrow \mathsf{isContr}(A).$$

Contruction. We construct $\mathsf{isContr}'$ by cofreely adding naturality to $\mathsf{isContr}$.

We define $\mathsf{isContr}'$ as a dependent presheaf over $\mathsf{Ty}_{\mathcal{C}}$. For any object $\Gamma : \mathcal{C}$ and element $A :: \mathsf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$, we let $\mathsf{isContr}'_{\Gamma}(A)$ be the set of functions c that send every morphism $\rho : \Delta \to \Gamma$ to a witness $c(\rho) : \mathsf{isContr}(A[\rho])$ of the contractibility of $A[\rho]$. For any morphism $f : \Omega \to \Gamma$, element $A :: \mathsf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$ and element $c : \mathsf{isContr}'_{\Gamma}(A)$, the restriction $c[f]$ sends a morphism $\rho : \Delta \to \Omega$ to a witness $c(f \circ \rho)$ of the contractibility of $A[f][\rho]$.

The map $\mathsf{isContr}'_{\Gamma}(A) \to \mathsf{isContr}(A)$ is defined by evaluating c at the identity morphism $\mathsf{id} : \Gamma \to \Gamma$. The map $\mathsf{isContr}(A) \to \mathsf{isContr}'_{\Gamma}(A)$ is defined using the fact that the weakly stable identity types are indeed weakly stable, providing maps $\mathsf{isContr}(A) \to \mathsf{isContr}(A[\rho])$ for any $\rho : \Delta \to \Gamma$. □

Theorem 5.12. *If $\mathcal{C}$ is equipped with weakly stable identity types, then the contractibility data constructed in construction 5.11 satisfies the assumptions of theorem 5.10.*

Proof. All of the assumptions of theorem 5.10 are standard properties of identity types. □

Remark 5.13. It should be possible to generalize this construction to CwFs without Σ-types by having the contractibility data quantify over telescopes of types, i.e. $\mathsf{isContr} : \mathsf{Ty}^{\star}_{\mathcal{C}} \to \mathcal{U}$.

6. Reflexive equivalences models

In this section, we construct $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs $\mathbf{PreReflGraph}(\mathcal{C})$ of pre-reflexive graphs, $\mathbf{PreReflEqv}(\mathcal{C})$ of equivalences with pre-reflexive equivalences and $\mathbf{ReflEqv}(\mathcal{C})$ of reflexive equivalences from a given $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ equipped with suitable contractibility data.

By *pre-reflexive* graph, we mean a graph together with a family of loops that should be thought of as reflexive loops, without conditions expressing the existence or the uniqueness of a reflexive loop yet.

For a SOGAT $\mathcal{T}$, the model $\mathbf{PreReflEqv}(\mathcal{T})$ will be used to prove external univalence for $\mathcal{T}$ by constructing identity types over $\mathcal{T}$ in section 7.

The $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathbf{PreReflGraph}(\mathcal{C})$ is an instance of an inverse diagram model (Kapulkin and Lumsdaine 2021), indexed by the inverse category

$$\mathsf{PreReflGraph} \triangleq \left\{ \begin{tikzcd} R \arrow[r, "e"] \arrow[rr, bend left, "c"] & E \arrow[r, shift left, "l"] \arrow[r, shift right, "r"'] & V \end{tikzcd} \right\},$$

where $l \circ e = c = r \circ e$, although we present it syntactically, rather than diagrammatically.

The inverse diagram PreReflGraph is an inverse replacement (Kraus and Sattler 2017) of the diagram

$$\mathsf{ReflGraph} \triangleq \left\{ \begin{tikzcd} E \arrow[r, bend left, "l"] \arrow[r, bend right, "r"'] & V \arrow[l, "e"'] \end{tikzcd} \right\}$$

that indexes the presheaf category of reflexive graphs.

The $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs $\mathbf{PreReflEqv}(\mathcal{C})$ of pre-reflexive equivalences and $\mathbf{ReflEqv}(\mathcal{C})$ of reflexive equivalences are homotopical inverse diagram models over the same base category PreReflGraph, with different sets of morphisms marked as equivalences. The types of $\mathbf{PreReflEqv}(\mathcal{C})$ and $\mathbf{ReflEqv}(\mathcal{C})$ will be types of $\mathbf{PreReflGraph}(\mathcal{C})$ along with some additional contractibility conditions for every marked arrow. For $\mathbf{PreReflEqv}(\mathcal{C})$, the arrows l and r are marked, while for $\mathbf{ReflEqv}(\mathcal{C})$, the arrows l, r and c are marked.

The CwF $\mathbf{ReflEqv}(\mathcal{C})$ of reflexive equivalences is essentially the same as the CwA of trivial auto-span-equivalences from Kapulkin and Lumsdaine (2018).

We fix a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ for the whole section.

6.1. The category of pre-reflexive graphs.

Definition 6.1. We define a category $\mathbf{PreReflGraph}(\mathcal{C})$ of **pre-reflexive graphs** in $\mathcal{C}$.

- An object Γ of $\mathbf{PreReflGraph}(\mathcal{C})$ is a triple $(\Gamma_V, \Gamma_E, \Gamma_R)$ where
$$\begin{aligned}\Gamma_V &:: \mathsf{Ob}_{\mathcal{C}},\\ \Gamma_E &:: \forall(\gamma_l, \gamma_r : \mathbf{y}(\Gamma_V)) \to \mathsf{Ty}_{\mathcal{C}},\\ \Gamma_R &:: \forall(\gamma : \Gamma)\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma, \gamma))) \to \mathsf{Ty}_{\mathcal{C}}.\end{aligned}$$
  We can see Γ_V as a type of vertices, Γ_E as a dependent type of edges and Γ_R as a dependent type of marked loops, which should be thought of as the reflexivity edges.
- A morphism f from Γ to Δ is a triple (f_V, f_E, f_R) where
$$\begin{aligned}&f_V :: \Gamma_V \to \Delta_V,\\ &f_E :: \forall(\gamma_l, \gamma_r : \mathbf{y}(\Gamma_V))\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma_l, \gamma_r))) \to \mathsf{Tm}_{\mathcal{C}}(\Delta_E(f_V(\gamma_l), f_V(\gamma_r))),\\ &f_R :: \forall(\gamma : \mathbf{y}(\Gamma_V))\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma, \gamma)))\ (\gamma_r : \mathsf{Tm}_{\mathcal{C}}(\Gamma_R(\gamma, \gamma_e))) \to \mathsf{Tm}_{\mathcal{C}}(\Delta_R(f_V(\gamma), f_E(\gamma_e))).\end{aligned}$$
- Identities and compositions are defined in the evident way. ⌟

We could instead define $\mathbf{PreReflGraph}(\mathcal{C})$ as the (non-equivalent) diagram category $\mathcal{C}^{\mathsf{PreReflGraph}}$. Using the more syntactic definition helps with computations in our applications.

6.2. **Reedy types.**

Definition 6.2. A **Reedy type** A, or **dependent pre-reflexive graph** A, over an object $\Gamma : \mathbf{PreReflGraph}(\mathcal{C})$ is a triple (A_V, A_E, A_R) where
$$\begin{aligned}A_V &:: \forall(\gamma : \mathbf{y}(\Gamma_V)) \to \mathsf{Ty}_{\mathcal{C}},\\ A_E &:: \forall\gamma_l\ \gamma_r\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma_l, \gamma_r)))\ (a_l : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma_l)))\ (a_r : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma_r))) \to \mathsf{Ty}_{\mathcal{C}},\\ A_R &:: \forall\gamma\ \gamma_e\ (\gamma_r : \mathsf{Tm}_{\mathcal{C}}(\Gamma_R(\gamma, \gamma_e)))\ (a : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma)))\ (a_e : \mathsf{Tm}_{\mathcal{C}}(A_E(\gamma_e, a, a))) \to \mathsf{Ty}_{\mathcal{C}}.\end{aligned}$$

The substitution of a Reedy-type A along a morphism $f : \Delta \to \Gamma$ in $\mathbf{PreReflGraph}(\mathcal{C})$ is defined by composition with the components of f:
$$\begin{aligned}A[f]_V &\triangleq \lambda\delta \mapsto A_V(f_V(\delta)),\\ A[f]_E &\triangleq \lambda\delta_e\ a_l\ a_r \mapsto A_E(f_E(\delta_e), a_l, a_r),\\ A[f]_R &\triangleq \lambda\delta_r\ a\ a_e \mapsto A_R(f_R(\delta_r, a, a_e)).\end{aligned}$$

The functoriality of this definition is easy to check; it follows from the associativity of function composition. ⌟

Definition 6.3. The representable Reedy types are defined in the same way: a **representable Reedy type** A, or **representable dependent pre-reflexive graph** A, over an object $\Gamma : \mathbf{PreReflGraph}(\mathcal{C})$ is a triple (A_V, A_E, A_R) where
$$\begin{aligned}A_V &:: \forall(\gamma : \mathbf{y}(\Gamma_V)) \to \mathsf{RepTy}_{\mathcal{C}},\\ A_E &:: \forall\gamma_l\ \gamma_r\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma_l, \gamma_r)))\ (a_l : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma_l)))\ (a_r : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma_r))) \to \mathsf{RepTy}_{\mathcal{C}},\\ A_R &:: \forall\gamma\ \gamma_e\ (\gamma_r : \Gamma_R(\gamma, \gamma_e))\ (a : \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma)))\ (a_e : \mathsf{Tm}_{\mathcal{C}}(A_E(\gamma_e, a, a))) \to \mathsf{RepTy}_{\mathcal{C}}.\end{aligned}$$
In particular, any representable Reedy type can be seen as a Reedy type by applying the map $\mathsf{RepTy}_{\mathcal{C}} \to \mathsf{Ty}_{\mathcal{C}}$ to all components. ⌟

Definition 6.4. A **term** of a Reedy type A over $\Gamma : \mathbf{PreReflGraph}(\mathcal{C})$ is a triple (a_V, a_E, a_R) where
$$\begin{aligned}a_V &:: \forall(\gamma : \mathbf{y}(\Gamma_V)) \to \mathsf{Tm}_{\mathcal{C}}(A_V(\gamma)),\\ a_E &:: \forall\gamma_l\ \gamma_r\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\Gamma_E(\gamma_l, \gamma_r))) \to \mathsf{Tm}_{\mathcal{C}}(A_E(\gamma_e, a_V(\gamma_l), a_V(\gamma_r))),\\ a_R &:: \forall\gamma\ \gamma_r\ (\gamma_r : \mathsf{Tm}_{\mathcal{C}}(\Gamma_R(\gamma, \gamma_e))) \to \mathsf{Tm}_{\mathcal{C}}(A_R(\gamma_r, a_V(\gamma), a_E(\gamma_e))).\end{aligned}$$

The substitution of a term of a Reedy type along a morphism in $\mathbf{PreReflGraph}(\mathcal{C})$ is also defined by composition with the components of f.

The extension of a context Γ by a type A is the context

$$\begin{aligned}
(\Gamma.A)_V &\triangleq \Gamma_V.A_V,\\
(\Gamma.A)_E((\gamma_l,a_l),(\gamma_r,a_r)) &\triangleq (\gamma_e : \Gamma_E(\gamma_l,\gamma_r)) \times (A_E(\gamma_e,a_l,a_r)),\\
(\Gamma.A)_R((\gamma,a),(\gamma_e,a_e)) &\triangleq (\gamma_r : \Gamma_R(\gamma,\gamma_e)) \times (A_R(\gamma_r,a,a_e)).
\end{aligned}$$

It can be checked that this definition satisfies the required universal property. ⌟

Construction 6.5. We equip the Reedy types and the representable Reedy types with **1**- and Σ- types as follows:

$$\begin{aligned}
\mathbf{1}_V &\triangleq \lambda\gamma \mapsto \mathbf{1},\\
\mathbf{1}_E &\triangleq \lambda\gamma_e\ t_l\ t_r \mapsto \mathbf{1},\\
\mathbf{1}_R &\triangleq \lambda\gamma_r\ t\ t_e \mapsto \mathbf{1},\\
(\Sigma(A,B))_V &\triangleq \lambda\gamma \mapsto (a : A_V(\gamma)) \times (b : B_V(\gamma,a))\\
(\Sigma(A,B))_E &\triangleq \lambda\gamma_e\ (a_l,b_l)\ (a_r,b_r) \mapsto (a_e : A_E(\gamma_e,a_l,a_r)) \times (b_e : B_E((\gamma_e,a_e),b_l,b_r))\\
(\Sigma(A,B))_R &\triangleq \lambda\gamma_r\ (a,b)\ (a_e,b_e) \mapsto (a_r : A_R(\gamma_r,a,a_e)) \times (b_r : B_r((\gamma_r,a_r),b,b_e)).
\end{aligned}$$

It is straightforward to check that these definitions are natural and satisfy the universal properties of **1**- and Σ- types. ⌟

Construction 6.6. We equip the Reedy types with Π-types with arities in the representable Reedy types as follows:

$$\begin{aligned}
(\Pi(A,B))_V &\triangleq \lambda\gamma \mapsto (a : A_V(\gamma)) \to B_V(\gamma,a)\\
(\Pi(A,B))_E &\triangleq \lambda\gamma_e\ f_l\ f_r \mapsto \forall a_l\ a_r\ (a_e : A_E(\gamma_e,a_l,a_r)) \to B_E((\gamma_e,a_e),f_l(a_l),f_r(a_r)),\\
(\Pi(A,B))_R &\triangleq \lambda\gamma_r\ f\ f_e \mapsto \forall a\ a_e\ (a_r : A_R(\gamma_r,a,a_e)) \to B_r((\gamma_r,a_r),f(a),f_e(a_e)).
\end{aligned}$$

Checking the naturality of this definition is straightfoward, and checking the universal property of the Π-types is a matter of unfolding the definitions. ⌟

To summarize, we have described the following construction.

Construction 6.7. If $\mathcal{C}$ is a $(\Sigma,\Pi_{\mathsf{rep}})$-CwF, then the category $\mathbf{PreReflGraph}(\mathcal{C})$ is equipped with the structure of a $(\Sigma,\Pi_{\mathsf{rep}})$-CwF whose types are the *Reedy types* (definition 6.2), and the projection functor $V : \mathbf{PreReflGraph}(\mathcal{C}) \to \mathcal{C}$ extends to a morphism of $(\Sigma,\Pi_{\mathsf{rep}})$-CwFs. ⌟

6.3. **Homotopical Reedy types.** Now assume that the $(\Sigma,\Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ is equipped with contractibility data, i.e. with families isContr and $\text{isContr}^{\mathsf{rep}}$ over its types and representable types, along with a map $\text{isContr}^{\mathsf{rep}}(A) \to \text{isContr}(A)$ for any $A : \mathsf{RepTy}_{\mathcal{C}}$.

Definition 6.8. A $\{l,r\}$**-homotopical Reedy type**, or **dependent pre-reflexive equivalence** is a Reedy type A over $\Gamma : \mathbf{PreReflGraph}(\mathcal{C})$ that satisfies the following two contractibility conditions:

$$\begin{aligned}
A.\overrightarrow{\mathsf{fun}} &: \forall\gamma_l\ \gamma_r\ (\gamma_e : \Gamma_E(\gamma_l,\gamma_r))\ (a_l : A_V(\gamma_l)) \to \text{isContr}((a_r : A_V(\gamma_r)) \times A_E(\gamma_e,a_l,a_r)),\\
A.\overleftarrow{\mathsf{fun}} &: \forall\gamma_l\ \gamma_r\ (\gamma_e : \Gamma_E(\gamma_l,\gamma_r))\ (a_r : A_V(\gamma_r)) \to \text{isContr}((a_l : A_V(\gamma_l)) \times A_E(\gamma_e,a_l,a_r)).
\end{aligned}$$

A representable dependent pre-reflexive equivalence is a representable Reedy type that satisfies:

$$\begin{aligned}
A.\overrightarrow{\mathsf{fun}} &: \forall\gamma_l\ \gamma_r\ (\gamma_e : \Gamma_E(\gamma_l,\gamma_r))\ (a_l : A_V(\gamma_l)) \to \text{isContr}^{\mathsf{rep}}((a_r : A_V(\gamma_r)) \times A_E(\gamma_e,a_l,a_r)),\\
A.\overleftarrow{\mathsf{fun}} &: \forall\gamma_l\ \gamma_r\ (\gamma_e : \Gamma_E(\gamma_l,\gamma_r))\ (a_r : A_V(\gamma_r)) \to \text{isContr}^{\mathsf{rep}}((a_l : A_V(\gamma_l)) \times A_E(\gamma_e,a_l,a_r)).
\end{aligned}$$

The action of morphism $f : \Delta \to \Gamma$ in **PreReflGraph** a on a dependent pre-reflexive equivalence A is defined by composition with the components of f. ⌟

In other words, a Reedy type is $\{l,r\}$-homotopical when it determines a dependent equivalence in the sense of definition 5.4.

Definition 6.9. A $\{l,r,c\}$**-homotopical Reedy type**, or **dependent reflexive equivalence** is a dependent pre-reflexive equivalence A over $\Gamma : \mathbf{PreReflGraph}(\mathcal{C})$ that satisfies the following additional contractibility conditions:

$$A.\mathsf{refl} : \forall \gamma\ (\gamma_e : \Gamma_E(\gamma,\gamma))\ (\gamma_r : \Gamma_R(\gamma,\gamma_e))\ a \to \text{isContr}((a_e : A_E(\gamma_e,a,a)) \times A_R(\gamma_r,a,a_e)).$$

A representable dependent reflexive equivalence is a representable dependent pre-reflexive equivalence that satisfies:

$$A.\mathsf{refl} : \forall \gamma\ (\gamma_e : \Gamma_E(\gamma,\gamma))\ (\gamma_r : \Gamma_R(\gamma,\gamma_e))\ a \to \text{isContr}^{\mathsf{rep}}((a_e : A_E(\gamma_e,a,a)) \times A_R(\gamma_r,a,a_e)).$$

We identify a collection of closure conditions on isContr$^{\mathsf{rep}}$ and isContr that ensure that the $\mathbf{1}$-, Σ- and Π-type formers of $\mathbf{PreReflGraph}(\mathcal{C})$ lift from the dependent pre-reflexive graphs to the dependent pre-reflexive equivalences.

Lemma 6.10. *Assume that the contractibility data is closed under the following operations:*

$$\begin{aligned}
\mathsf{contr}^{\mathsf{rep}}_{\mathbf{1}}\ &: \text{isContr}^{\mathsf{rep}}(\mathbf{1}\times\mathbf{1}),\\
\mathsf{contr}^{\mathsf{rep}}_{\Sigma}\ &: \forall (A : \mathsf{RepTy}_{\mathcal{C}})\ (B : \forall a \to \mathsf{RepTy}_{\mathcal{C}})\ (C : \forall a \to \mathsf{RepTy}_{\mathcal{C}})\ (D : \forall a\ b\ c \to \mathsf{RepTy}_{\mathcal{C}})\\
&\qquad \to \text{isContr}^{\mathsf{rep}}((a : A) \times B(a))\\
&\qquad \to (\forall a\ b \to \text{isContr}^{\mathsf{rep}}((c : C(a)) \times D(a,b,c)))\\
&\qquad \to \text{isContr}^{\mathsf{rep}}(((a : A) \times (c : C(a))) \times ((b : B(a)) \times D(a,b,c))),\\
\mathsf{contr}^{\mathsf{rep}}_{\cong}\ &: \forall (A,B : \mathsf{RepTy}_{\mathcal{C}})\\
&\qquad \to (\mathsf{Tm}_{\mathcal{C}}(A) \cong \mathsf{Tm}_{\mathcal{C}}(B)) \to \text{isContr}^{\mathsf{rep}}(A)\\
&\qquad \to \text{isContr}^{\mathsf{rep}}(B),\\
\mathsf{contr}_{\mathbf{1}}\ &: \text{isContr}(\mathbf{1}\times\mathbf{1}),\\
\mathsf{contr}_{\Sigma}\ &: \forall (A : \mathsf{Ty}_{\mathcal{C}})\ (B : \forall a \to \mathsf{Ty}_{\mathcal{C}})\ (C : \forall a \to \mathsf{Ty}_{\mathcal{C}})\ (D : \forall a\ b\ c \to \mathsf{Ty}_{\mathcal{C}})\\
&\qquad \to \text{isContr}((a : A) \times B(a))\\
&\qquad \to (\forall a\ b \to \text{isContr}((c : C(a)) \times D(a,b,c)))\\
&\qquad \to \text{isContr}(((a : A) \times (c : C(a))) \times ((b : B(a)) \times D(a,b,c))),\\
\mathsf{contr}_{\cong}\ &: \forall (A,B : \mathsf{Ty}_{\mathcal{C}})\\
&\qquad \to (\mathsf{Tm}_{\mathcal{C}}(A) \cong \mathsf{Tm}_{\mathcal{C}}(B)) \to \text{isContr}(A)\\
&\qquad \to \text{isContr}(B),\\
\mathsf{contr}_{\Pi}\ &: \forall (X : \mathsf{RepTy}_{\mathcal{C}})\ (Y : \mathsf{Tm}_{\mathcal{C}}(X) \to \mathsf{RepTy}_{\mathcal{C}})\\
&\quad \to (\forall x \to \text{isContr}^{\mathsf{rep}}(Y(x)))\\
&\quad \to (\underline{A} : \mathsf{Tm}_{\mathcal{C}}(X) \to \mathsf{Ty}_{\mathcal{C}})\\
&\quad \to (\underline{B} : (x : \mathsf{Tm}_{\mathcal{C}}(X)) \to \mathsf{Tm}_{\mathcal{C}}(Y(x)) \to \mathsf{Tm}_{\mathcal{C}}(\underline{A}(x)) \to \mathsf{Ty}_{\mathcal{C}})\\
&\quad \to (\forall x\ y \to \text{isContr}((a : \underline{A}(x)) \times \underline{B}(x,y,a)))\\
&\quad \to \text{isContr}((a : (x : X) \to \underline{A}(x))\\
&\qquad\qquad \times ((x : X) \to (y : Y(x)) \to \underline{B}(x,y,a(x)))).
\end{aligned}$$
⌟

Then the $\mathbf{1}$-, Σ- *and* Π- *type structures lift from* $\mathbf{PreReflGraph}(\mathcal{C})$ *to* $\mathbf{PreReflEqv}(\mathcal{C})$*, for both types and representable types.*

Proof. We need to check two contractibility conditions for each type former.

Case 1: We have to check the following two contractibility conditions:

$$\begin{aligned}
&\forall \gamma_l\ \gamma_r\ \gamma_e\ t_l \to \text{isContr}(\mathbf{1}\times\mathbf{1}),\\
&\forall \gamma_l\ \gamma_r\ \gamma_e\ t_r \to \text{isContr}(\mathbf{1}\times\mathbf{1}).
\end{aligned}$$

They are both instances of $\mathsf{contr}_{\mathbf{1}}$, or $\mathsf{contr}^{\mathsf{rep}}_{\mathbf{1}}$ in the case of representable Reedy types.

Case Σ: We have a dependent pre-reflexive equivalence A over Γ and a dependent pre-reflexive equivalence B over $(\Gamma.A)$.

In order to check that the Reedy type $\Sigma(A,B)$ is a dependent pre-reflexive equivalence, we have to check the following two contractibility conditions:

$$\begin{aligned}
&\forall \gamma_l\ \gamma_r\ \gamma_e\ (a_l,b_l) \to \mathrm{isContr}(((a_r : A_V(\gamma_r)) \times (b_r : B_V(\gamma_r,a_r))) \\
&\qquad\qquad \times ((a_e : A_E(\gamma_e,a_l,a_r)) \times (b_e : B_E((\gamma_e,a_e),b_l,b_r)))), \\
&\forall \gamma_l\ \gamma_r\ \gamma_e\ (a_r,b_r) \to \mathrm{isContr}(((a_l : A_V(\gamma_l)) \times (b_l : B_V(\gamma_l,a_l))) \\
&\qquad\qquad \times ((a_e : A_E(\gamma_e,a_l,a_r)) \times (b_e : B_E((\gamma_e,a_e),b_l,b_r)))).
\end{aligned}$$

They both follow from $\mathrm{isContr}_\Sigma$ and from the contractibility conditions of A and B.
In the case of representable Reedy types, we use $\mathrm{isContr}^{\mathsf{rep}}_\Sigma$ instead.

Case Π: We have a representable dependent pre-reflexive equivalence A over Γ and a dependent pre-reflexive equivalence B over $(\Gamma.A)$.

In order to check that the Reedy type $\Pi(A,B)$ is a dependent pre-reflexive equivalence over Γ, we have to check the following two contractibility conditions:

$$\begin{aligned}
&\forall \gamma_l\ \gamma_r\ \gamma_e\ f_l \to \mathrm{isContr}((f_r : \forall a_r \to B_V(\gamma_r,a_r)) \\
&\qquad\qquad \times (f_e : \forall a_l\ a_r\ a_e \to B_E((\gamma_e,a_e),f_l(a_l),f_r(a_r)))), \\
&\forall \gamma_l\ \gamma_r\ \gamma_e\ f_r \to \mathrm{isContr}((f_l : \forall a_l \to B_V(\gamma_l,a_l)) \\
&\qquad\qquad \times (f_e : \forall a_l\ a_r\ a_e \to B_E((\gamma_e,a_e),f_l(a_l),f_r(a_r)))).
\end{aligned}$$

Up to type isomorphism, the first contractibility condition is an instance of contr_Π, whose arguments $(X,Y,\underline{A},\underline{B})$ are instantiated to:

$$\begin{aligned}
X &= A_V(\gamma_r), \\
Y(a_r) &= (a_l : A_V(\gamma_l)) \times (a_e : A_E(\gamma_e,a_l,a_r)), \\
\underline{A}(a_r) &= B_V(\gamma_r,a_r), \\
\underline{B}(a_r,(a_l,a_e),b_r) &= B_E((\gamma_e,a_e),f_l(a_l),f_r(a_r)).
\end{aligned}$$

Relying on type isomorphisms is allowed thanks to the operations $\mathsf{contr}^{\mathsf{rep}}_{\cong}$ and $\mathsf{contr}_{\cong}$.
Up to symmetry, the second contractibility condition is similar the first one. □

Construction 6.11. Let $\mathcal{C}$ be a $(\Sigma,\Pi_{\mathsf{rep}})$-CwF equipped with operations satisfying the specification of lemma 6.10. Then there is a $(\Sigma,\Pi_{\mathsf{rep}})$-CwF $\mathbf{PreReflEqv}(\mathcal{C})$ whose types are the dependent pre-reflexive equivalences as defined in definition 6.8. There is a $(\Sigma,\Pi_{\mathsf{rep}})$-CwF $\mathbf{PreReflEqv}(\mathcal{C}) \to \mathbf{PreReflGraph}(\mathcal{C})$ lying over the identity functor that forgets the contractibility witnesses of the dependent pre-reflexive equivalences. ⌟

Lemma 6.12. *Assume that the contractibility data is closed under the operations of lemma 6.10 and the additional operation*

$$\begin{aligned}
\mathsf{contr}_{\Pi,\mathsf{refl}} : {} & \forall (X : \mathsf{RepTy}_{\mathcal{C}})\ (Y : \mathsf{Tm}_{\mathcal{C}}(X) \to \mathsf{RepTy}_{\mathcal{C}})\ ((Z : \mathsf{Tm}_{\mathcal{C}}(X) \to \mathsf{RepTy}_{\mathcal{C}})) \\
& \to (\forall x \to \mathrm{isContr}^{\mathsf{rep}}(Y(x))) \\
& \to (\forall x \to \mathrm{isContr}^{\mathsf{rep}}(Z(x))) \\
& \to (f : \forall x \to \mathsf{Tm}_{\mathcal{C}}(Y(x)) \to \mathsf{Tm}_{\mathcal{C}}(Z(x))) \\
& \to (\underline{A} : (x : \mathsf{Tm}_{\mathcal{C}}(X)) \to \mathsf{Tm}_{\mathcal{C}}(Z(x)) \to \mathsf{Ty}_{\mathcal{C}}) \\
& \to (\underline{B} : (x : \mathsf{Tm}_{\mathcal{C}}(X)) \to (y : \mathsf{Tm}_{\mathcal{C}}(Y(x))) \to \mathsf{Tm}_{\mathcal{C}}(\underline{A}(x,f(x,y))) \to \mathsf{Ty}_{\mathcal{C}}) \\
& \to (\forall x\ y \to \mathrm{isContr}((a : \underline{A}(x,f(x,y))) \times \underline{B}(x,y,a))) \\
& \to \mathrm{isContr}((a : (x : X) \to (z : Z(x)) \to \underline{A}(x,z)) \\
& \qquad\qquad \times ((x : X) \to (y : Y(x)) \to \underline{B}(x,y,a(x,f(y))))).
\end{aligned}$$

⌟

Then the $\mathbf{1}$-, Σ- *and* Π- *type structures lift from* $\mathbf{PreReflEqv}(\mathcal{C})$ *to* $\mathbf{ReflEqv}(\mathcal{C})$*, for both types and representable types.*

Proof. We need to check one contractibility condition for each type former.

Case 1: We have to check the following contractibility condition:

$$\forall \gamma\ \gamma_e\ \gamma_r\ t \to \mathrm{isContr}(\mathbf{1} \times \mathbf{1}).$$

This is an instance of $\mathsf{contr}_{\mathbf{1}}$, or $\mathsf{contr}^{\mathsf{rep}}_{\mathbf{1}}$ in the case of representable Reedy types.

Case Σ: We have a dependent reflexive equivalence A over Γ and a dependent reflexive equivalence B over $(\Gamma.A)$.

In order to check that the Reedy type $\Sigma(A, B)$ is a dependent reflexive equivalence, we have to check the following contractibility condition:

$$\begin{aligned}\forall \gamma\ \gamma_e\ \gamma_r\ (a, b) \to \mathrm{isContr}(((a_e : A_E(\gamma_e, a, a)) \times (b_e : B_E((\gamma_e, a_e), b, b)))\\ \times ((a_r : A_R(\gamma_r, a, a_e)) \times (b_r : B_R((\gamma_r, a_r), b, b_e)))).\end{aligned}$$

It follows from $\mathrm{isContr}_\Sigma$ (or $\mathrm{isContr}^{\mathsf{rep}}_\Sigma$) and from the contractibility conditions of A and B.

Case Π: We have a representable dependent reflexive equivalence A over Γ and a dependent reflexive equivalence B over $(\Gamma.A)$.

In order to check that the Reedy type $\Pi(A, B)$ is a dependent reflexive equivalence, we have to check the following contractibility condition:

$$\begin{aligned}\forall \gamma\ \gamma_e\ \gamma_r\ f \to \mathrm{isContr}((f_e : \forall a_l\ a_r\ (a_e : A_E(\gamma_e, a_l, a_r)) \to B_E((\gamma_e, a_e), f(a_l), f(a_r)))\\ \times (f_r : \forall a\ a_e\ (a_r : A_R(\gamma_r, a, a_e)) \to B_R((\gamma_r, a_r), f(a), f_e(a_e)))).\end{aligned}$$

Up to type isomorphism, this contractibility condition is an instance of $\mathsf{contr}_{\Pi,\mathsf{refl}}$, whose arguments $(X, Y, Z, f, \underline{A}, \underline{B})$ are instantiated to:

$$\begin{aligned}
X &= A_V(\gamma),\\
Y(a) &= (a_e : A_V(\gamma_e, a, a)) \times (a_r : A_R(\gamma_r, a, a_e)),\\
Z(a) &= (a_r : A_V(\gamma)) \times (a_e : A_E(\gamma_e, a, a_r)),\\
f(a, (a_e, a_r)) &= (a, a_e),\\
\underline{A}(a, (a_r, a_e)) &= B_E((\gamma_e, a_e), f(a), f(a_r)),\\
\underline{B}(a, (a_e, a_r), b_r) &= B_R((\gamma_r, a_r), f(a), f_e(a_e)).
\end{aligned}$$

□

Construction 6.13. Let $\mathcal{C}$ be a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF equipped with operations satisfying the specifications of lemma 6.10 and lemma 6.12. Then there is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathbf{ReflEqv}(\mathcal{C})$ whose types are the dependent reflexive equivalences as defined in definition 6.8. There is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathbf{ReflEqv}(\mathcal{C}) \to \mathbf{PreReflEqv}(\mathcal{C})$ lying over the identity functor that forgets the additional contractibility conditions of dependent reflexive equivalences. ⌟

6.4. Parametricity structures.

We now use the pre-reflexive graph and homotopical pre-reflexive graphs models to specify notions of parametricity structures over $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs.

Definition 6.14. A **parametricity structure** for a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ is a section $[\![-]\!]$ of the projection morphism $V : \mathbf{PreReflGraph}(\mathcal{C}) \to \mathcal{C}$.

A $\{l, r\}$**-homotopical parametricity structure** for a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ that is equipped with contractibility data and satisfies the closure conditions of lemma 6.10 is a section $[\![-]\!]$ of the projection morphism V : $\mathbf{PreReflEqv}(\mathcal{C}) \to \mathcal{C}$. ⌟

Definition 6.15. Let $\mathcal{C}$ be a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF equipped with a parametricity structure $[\![-]\!]$.

A **reflexivity operation** for an object $\Gamma : \mathcal{C}$ consists of:

$$\begin{aligned}
\mathsf{refl}^E_\Gamma &:: \forall(\gamma : \mathrm{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}([\![\Gamma]\!]_E(\gamma, \gamma)),\\
\mathsf{refl}^R_\Gamma &:: \forall(\gamma : \mathrm{y}(\Gamma)) \to \mathsf{Tm}_{\mathcal{C}}([\![\Gamma]\!]_R(\gamma, \mathsf{refl}^E_\Gamma(\gamma))).
\end{aligned}$$

A **reflexivity operation** for a type $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$ consists of:

$$\begin{aligned}\mathsf{refl}^E_A &:: \forall (\gamma : \mathbf{y}(\Gamma))\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\llbracket\Gamma\rrbracket_E(\gamma,\gamma)))\ (\gamma_r : \mathsf{Tm}_{\mathcal{C}}(\llbracket\Gamma\rrbracket_R(\gamma,\gamma_e)))\\ &\qquad (a : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Tm}_{\mathcal{C}}(\llbracket A\rrbracket_E(\gamma_e, a, a)),\\ \mathsf{refl}^R_A &:: \forall (\gamma : \mathbf{y}(\Gamma))\ (\gamma_e : \mathsf{Tm}_{\mathcal{C}}(\llbracket\Gamma\rrbracket_E(\gamma,\gamma)))\ (\gamma_r : \mathsf{Tm}_{\mathcal{C}}(\llbracket\Gamma\rrbracket_R(\gamma,\gamma_e)))\\ &\qquad (a : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Tm}_{\mathcal{C}}(\llbracket A\rrbracket_R(\gamma_r, a, \mathsf{refl}^E_A(\gamma_r, a))).\end{aligned}$$

A **reflexivity structure** over $\mathcal{C}$ consists of reflexivity operations for all objects and types of $\mathcal{C}$. ⌟

Proposition 6.16. *If $\mathcal{C}$ is a contextual $(\Sigma, \Pi_{\mathsf{rep}})$-CwF that is equipped with reflexivity operations for all types, then it is also equipped with reflexivity operations for all objects, and thus of a reflexivity structure.*

Proof. By induction on the contexts of $\mathcal{C}$, we pose

$$\begin{aligned}\mathsf{refl}^E_\diamond(\star) &\triangleq \star,\\ \mathsf{refl}^R_\diamond(\star) &\triangleq \star,\\ \mathsf{refl}^E_{\Gamma.A}(\gamma, a) &\triangleq (\mathsf{refl}^E_\Gamma(\gamma), \mathsf{refl}^E_A(\gamma, \mathsf{refl}^E_\Gamma(\gamma), \mathsf{refl}^R_\Gamma(\gamma), a)),\\ \mathsf{refl}^R_{\Gamma.A}(\gamma, a) &\triangleq (\mathsf{refl}^R_\Gamma(\gamma), \mathsf{refl}^R_A(\gamma, \mathsf{refl}^E_\Gamma(\gamma), \mathsf{refl}^R_\Gamma(\gamma), a)).\end{aligned}$$ □

We now fix a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{C}$ equipped with contractibility data and with a homotopical parametricity structure $\llbracket-\rrbracket$, along with a reflexivity structure.

We can then equip it with reflexive equivalences (definition 5.3) and with dependent equivalences (definition 5.4):

$$\begin{aligned}\mathsf{Id}_A(\gamma, a_l, a_r) &\triangleq \llbracket A\rrbracket(\mathsf{refl}^E_\Gamma(\gamma), a_l, a_r),\\ \mathsf{refl}_A(\gamma, a) &\triangleq \mathsf{refl}^E_A(\mathsf{refl}^E_\Gamma(\gamma), \mathsf{refl}^R_\Gamma(\gamma), a),\\ \mathsf{DId}_{A.B}(\gamma, a_l, a_r, a_e, b_l, b_r) &\triangleq \llbracket B\rrbracket((\mathsf{refl}^E_\Gamma(\gamma), a_e), b_l, b_r).\end{aligned}$$

Our goal is now to investigate the remaining assumption of theorem 5.10. We prove some lemmata showing that center and homogeneous all-paths operations are preserved by the operations of lemma 6.10, under some additional hypothesis for some of the operations. These lemmata will be needed in section 7.

Construction 6.17. Let $\mathsf{Ty}'_{\mathcal{C}} \hookrightarrow \mathsf{Ty}_{\mathcal{C}}$ be a subfamily of the family of types such that:

- for every $X :: \mathbf{y}(\Gamma) \to \mathsf{Ty}'_{\mathcal{C}}$, the contractibility witnesses of the homotomical Reedy type $\llbracket X\rrbracket$ are equipped with centers.
- for every $X :: \mathbf{y}(\Gamma) \to \mathsf{Ty}'_{\mathcal{C}}$, the dependent types $\llbracket X\rrbracket_E$ and $\llbracket X\rrbracket_R$ are dependent types in $\mathsf{Ty}'_{\mathcal{C}}$, i.e. they factor through $\mathsf{Ty}'_{\mathcal{C}} \hookrightarrow \mathsf{Ty}_{\mathcal{C}}$.

Let $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}'_{\mathcal{C}}$ be a global type and $B :: (\gamma : \mathbf{y}(\Gamma)) \to (a : \mathsf{Tm}_{\mathcal{C}}(A(\gamma))) \to \mathsf{Ty}'_{\mathcal{C}}$ be a global dependent types. If we have a homogeneous all-paths operation for B over $\Gamma.A$, then we construct a heterogeneous all-paths operation for B.

Proof. We have to define

$$\begin{aligned}B.\mathsf{path} &:: \forall \gamma\ (a_l, a_r : \mathsf{Tm}_{\mathcal{C}}(A(\gamma)))\ (a_e : \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma, a_l, a_r)))\\ &\qquad (b_l : \mathsf{Tm}_{\mathcal{C}}(B(\gamma, a_l)))\ (b_r : \mathsf{Tm}_{\mathcal{C}}(B(\gamma, a_r))) \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{DId}_{A.B}(\gamma, a_e, b_l, b_r)).\end{aligned}$$

We first transport b_r through the equivalence $\mathsf{DId}_{A.B}(\gamma, a_e) : B(\gamma, a_l) \simeq B(\gamma, a_r)$.

$$\begin{aligned}\overleftarrow{T}_B(\gamma, a_e, b_r) &: \ \mathsf{isContr}((b'_l : B(\gamma, a_l)) \times (b'_e : \mathsf{DId}_{A.B}(\gamma, a_e, b'_l, b_r))),\\ \overleftarrow{T}_B(\gamma, a_e, b_r) &\triangleq \mathsf{DId}_{A.B}(\gamma, a_e).\overleftarrow{\mathsf{fun}}(b_r),\\ (b'_l(\gamma, a_e, b_r), b'_e(\gamma, a_e, b_r)) &\triangleq \overleftarrow{T}_B.\mathsf{center}(\gamma, a_e, b_r),\end{aligned}$$

where the center can be obtained thanks to our first assumption about $\mathsf{Ty}'_{\mathcal{C}}$.

Now, using the homogeneous all-paths operation for B at a_l, we obtain a homogeneous path between b_l and b'_l.

$$\begin{aligned} b''_e(\gamma, a_e, b_l, b_r) &: \ \mathsf{Id}_{B(\gamma,a_l)}(\gamma, b_l, b'_l), \\ b''_e(\gamma, a_e, b_l, b_r) &\triangleq B.\mathsf{hpath}((\gamma, a_l), b_l, b'_l). \end{aligned}$$

It remains to compose the homogeneous path b''_e with the homogeneous path b'_e.

$$\begin{aligned} I_B(\gamma, a_e, b_r)(b_l) \quad &\triangleq \mathsf{DId}_{A.B}(\gamma, a_e, b_l, b_r), \\ T_I(\gamma, a_e, b_l, b_r) \quad &: \ I_B(\gamma, a_e, b_r)(b_l) \simeq I_B(\gamma, a_e, b_r)(b'_l), \\ T_I(\gamma, a_e, b_l, b_r) \quad &\triangleq \mathsf{DId}_{B.I_B}((\gamma, a_e, b_r), b''_e), \\ \overleftarrow{T_I}(\gamma, a_e, b_l, b_r) \quad &: \ \text{isContr}((b_e : I_B(\gamma, a_e, b_r)(b_l)) \times \cdots), \\ \overleftarrow{T_I}(\gamma, a_e, b_l, b_r) \quad &\triangleq T_I(\gamma, a_e, b_l, b_r).\overleftarrow{\mathsf{fun}}(b''_e(\gamma, a_e, b_l, b_r)), \\ (b_e(\gamma, a_e, b_l, b_r), \_) &\triangleq \overleftarrow{T_I}.\mathsf{center}(\gamma, a_e, b_l, b_r), \end{aligned}$$

where the center can be obtained thanks to our first assumption about $\mathsf{Ty}'_{\mathcal{C}}$ and the fact that the dependent type I_B lands in $\mathsf{Ty}'_{\mathcal{C}}$.

We can finally pose:

$$B.\mathsf{path}(\gamma, a_e, b_l, b_r) \triangleq b_e(\gamma, a_e, b_l, b_r). \qquad \square$$

Construction 6.18. Let $A, B :: \text{よ}(\Gamma) \to \mathsf{Ty}_{\mathcal{C}}$ be two types related by an isomorphism $\alpha : \forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma)) \cong \mathsf{Tm}_{\mathcal{C}}(B(\gamma))$.

If A is equipped with a center operation (resp. with a homogeneous all-paths operations), then we equip B with a center operation (resp. with a homogeneous all-paths operations).

Proof. Equipping B with a center operation is straightforward:

$$B.\mathsf{center}(\gamma) \triangleq \alpha(\gamma, A.\mathsf{center}(\gamma));$$

For the homogeneous all-paths operations, we show that the isomorphism α lifts an isomorphism between Id_A and Id_B.

We have $[\![\alpha]\!]_E : \forall \gamma_l\ \gamma_r\ \gamma_e\ a_l\ a_r \to \mathsf{Tm}_{\mathcal{C}}([\![A]\!]_E(\gamma_e, a_l, a_r)) \cong \mathsf{Tm}_{\mathcal{C}}([\![B]\!]_E(\gamma_e, \alpha(\gamma_l, a_l), \alpha(\gamma_r, a_r)))$. Thus $[\![\alpha]\!]_E(\mathsf{refl}^E_\Gamma(\gamma)) : \forall a_l\ a_r \to \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_A(\gamma, a_l, a_r)) \cong \mathsf{Tm}_{\mathcal{C}}(\mathsf{Id}_B(\gamma, \alpha(\gamma, a_l), \alpha(\gamma, a_r)))$ is an isomorphism between Id_A and Id_B.

We can now pose

$$B.\mathsf{hpath}(\gamma, b_l, b_r) \triangleq [\![\alpha_E]\!](\mathsf{refl}^E_\Gamma(\gamma), A.\mathsf{hpath}(\gamma, \alpha^{-1}(\gamma, b_l), \alpha^{-1}(\gamma, b_r))). \qquad \square$$

Construction 6.19. The type $\mathbf{1} \times \mathbf{1}$ has center and homogeneous all-paths operations over any context $\Gamma : \mathcal{C}$.

Proof. The center is $(\mathrm{tt}, \mathrm{tt})$ over any context.

By definition of $\mathbf{1}$- and Σ- types in $\mathbf{PreReflGraph}(\mathcal{C})$, we compute $\mathsf{Id}_{\mathbf{1}\times\mathbf{1}}(\gamma, -, -) = [\![\mathbf{1} \times \mathbf{1}]\!]_E(\cdots) = \mathbf{1} \times \mathbf{1}$. Thus the homogeneous all-paths operations can also be defined by $(\mathrm{tt}, \mathrm{tt})$ over any context. $\square$

Construction 6.20. Assume given the data of:

$$\begin{aligned} &\Gamma \ : \ \mathcal{C}, \\ &A :: \forall \gamma \to \mathsf{Ty}_{\mathcal{C}}, \\ &B :: \forall \gamma\ a \to \mathsf{Ty}_{\mathcal{C}}, \\ &C :: \forall \gamma\ a \to \mathsf{Ty}_{\mathcal{C}}, \\ &D :: \forall \gamma\ a\ b\ c \to \mathsf{Ty}_{\mathcal{C}}, \end{aligned}$$

along with center operations for the types

$$(a : A(\gamma)) \times B(\gamma, a)$$

over $(\gamma : \mathbb{y}(\Gamma))$ and

$$(c : C(\gamma, a)) \times D(\gamma, a, b, c)$$

over $((\gamma, a, b) : \mathbb{y}(\Gamma.A.B))$.

We construct an center over $(\gamma : \mathbb{y}(\Gamma))$ for the type

$$((a : A(\gamma)) \times (b : B(\gamma, a))) \times ((c : C(\gamma, a)) \times (d : D(\gamma, a, b, c))).$$

Construction. We write $AB \triangleq (a : A(\gamma)) \times B(\gamma, a)$, etc.

From our hypotheses, we have

$$\begin{aligned}\langle a, b\rangle(\gamma) &\triangleq \mathsf{center}_{AB}(\gamma),\\ \langle c, d\rangle(\gamma) &\triangleq \mathsf{center}_{CD}(\gamma, ab(\gamma)).\end{aligned}$$

Thus we can pose

$$\mathsf{center}_{ABCD}(\gamma) \triangleq ((a(\gamma), c(\gamma)), (b(\gamma), d(\gamma))).$$ □

Construction 6.21. Assume given the data of:

$$\begin{aligned}\Gamma &: \mathcal{C},\\ A &:: \forall \gamma \to \mathsf{Ty}_{\mathcal{C}},\\ B &:: \forall \gamma\ a \to \mathsf{Ty}_{\mathcal{C}},\\ C &:: \forall \gamma\ a \to \mathsf{Ty}_{\mathcal{C}},\\ D &:: \forall \gamma\ a\ b\ c \to \mathsf{Ty}_{\mathcal{C}},\end{aligned}$$

along with a homogeneous all-paths operation for the type

$$(a : A(\gamma)) \times B(\gamma, a)$$

over $(\gamma : \mathbb{y}(\Gamma))$ and a heterogeneous all-paths operation for the dependent type

$$(a, b) \mapsto (c : C(\gamma, a)) \times D(\gamma, a, b, c)$$

over $(\gamma : \mathbb{y}(\Gamma))$.

We construct a homogeneous all-paths operation over $(\gamma : \mathbb{y}(\Gamma))$ for the type

$$((a : A(\gamma)) \times (b : B(\gamma, a))) \times ((c : C(\gamma, a)) \times (d : D(\gamma, a, b, c))).$$

Construction. We pose $AB \triangleq (a : A(\gamma)) \times B(\gamma, a)$, etc. We also write ab instead of (a, b), etc.

Our goal is to define

$$\mathsf{hpath}_{ABCD} : \forall \gamma\ acbd_l\ acbd_r \to \mathsf{Id}_{ACBD}(acbd_l, acbd_r).$$

We pose

$$\begin{aligned}ab_e(\gamma, ab_l, ab_r) &\triangleq \mathsf{hpath}_{AB}(\gamma, ab_l, ab_r),\\ cd_e(\gamma, acbd_l, acbd_r) &\triangleq \mathsf{path}_{CD}(\gamma, ab_e, cd_l, cd_r).\end{aligned}$$

Thus, we can define

$$\mathsf{hpath}_{ABCD}(\gamma, acbd_l, acbd_r) \triangleq acbd_e(\gamma, acbd_l, acbd_r).$$ □

Construction 6.22. Also assume given the data of:

$$\begin{aligned}\Gamma &: \mathcal{C},\\ X &:: \mathbb{y}(\Gamma) \to \mathsf{RepTy}_{\mathcal{C}},\\ Y &:: \forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(X(\gamma)) \to \mathsf{RepTy}_{\mathcal{C}},\end{aligned}$$

$$\begin{aligned}
&c_Y :: \forall \gamma\ x \to \text{isContr}^{\mathsf{rep}}(Y(\gamma, x)),\\
&A\ :: \forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(X(\gamma)) \to \mathsf{Ty}_{\mathcal{C}},\\
&B\ :: \forall \gamma\ (x : \mathsf{Tm}_{\mathcal{C}}(X(\gamma)))\ (y : \mathsf{Tm}_{\mathcal{C}}(Y(\gamma, x))) \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma, x)) \to \mathsf{Ty}_{\mathcal{C}},
\end{aligned}$$

along with a center operation for the type

$$AB(\gamma) \triangleq (a : A(\gamma, x)) \times B(\gamma, x, y, a)$$

over $((\gamma, x, y) : \mathbb{y}(\Gamma.X.Y))$.

We construct an center over $(\gamma : \mathbb{y}(\Gamma))$ for the type

$$(a : (x : X(\gamma)) \to A(\gamma, x)) \times (b : (x : X(\gamma)) \to (y : Y(\gamma, x)) \to B(\gamma, x, y, a(x))).$$

Construction. Our goal is to define the following:

$$\begin{aligned}
\mathsf{center}_{\Pi}(\gamma) : (&a : (x : X(\gamma)) \to A(\gamma, x))\\
&\times (b : (x : X(\gamma))\ (y : Y(\gamma, x)) \to B(\gamma, x, y, a(x))).
\end{aligned}$$

Since Y is a family of contractible representable types, we can find its centers of contraction.

$$y_0(\gamma, x) \triangleq \mathsf{center}_Y(\gamma, x),$$

We then obtain elements of A from the centers of contraction of AB.

$$\begin{aligned}
&a(\gamma, x) &&: \ A(\gamma, x),\\
&b_0(\gamma, x) &&: \ B(\gamma, x, y_0(\gamma, x), a(\gamma, x)),\\
&(a(\gamma, x), b_0(\gamma, x)) \triangleq \mathsf{center}_{AB}(\gamma, x, y_0(\gamma, x)).
\end{aligned}$$

We then want to transport b_0 over paths in Y, using the fact that Y has all paths. We start by computing paths from y_0 to any element in Y.

$$\begin{aligned}
&p_Y(\gamma, x, y) : \ \mathsf{Id}_{Y[x]}(\gamma, y_0(x), y),\\
&p_Y(\gamma, x, y) \triangleq \mathsf{hpath}_Y((\gamma, x), y_0(\gamma, x), y).
\end{aligned}$$

We now consider the transport of b_0 over p_Y.

$$\begin{aligned}
&T_B(\gamma, x, y) : \ B(\gamma, x, y_0(\gamma, x), a(\gamma, x)) \simeq B(\gamma, x, y, a(\gamma, x)),\\
&T_B(\gamma, x, y) \triangleq \mathsf{DId}_{(y:Y).B(\gamma,x,y,a(\gamma,x))}((\gamma, x), p_Y(\gamma, x, y)),\\
&\overrightarrow{T_B}(\gamma, x, y) : \ \text{isContr}((b : B(\gamma, x, y, a(\gamma, x))) \times \cdots),\\
&\overrightarrow{T_B}(\gamma, x, y) \triangleq T_B(\gamma, x, y).\overrightarrow{\mathsf{fun}}(b_0(\gamma, x)),\\
&b(\gamma, x, y) \quad \triangleq \overrightarrow{T_B}.\mathsf{center}(\gamma, x, y).
\end{aligned}$$

We can now conclude the definition:

$$\mathsf{center}_{\Pi}(\gamma) \triangleq (\lambda x \mapsto a(\gamma, x), \lambda x\ y \mapsto b(\gamma, x, y)).$$ □

Construction 6.23. Assume that the weakly stable identity type introduction structure Id_{-} on $\mathsf{RepTy}_{\mathcal{C}}$ can be equipped with a weakly stable elimination structure.

Also given the data of:

$$\begin{aligned}
&\Gamma\ :\ \mathcal{C},\\
&X :: \mathbb{y}(\Gamma) \to \mathsf{RepTy}_{\mathcal{C}},\\
&Y :: \forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(X(\gamma)) \to \mathsf{RepTy}_{\mathcal{C}},\\
&A :: \forall \gamma \to \mathsf{Tm}_{\mathcal{C}}(X(\gamma)) \to \mathsf{Ty}_{\mathcal{C}},\\
&B :: \forall \gamma\ (x : \mathsf{Tm}_{\mathcal{C}}(X(\gamma)))\ (y : \mathsf{Tm}_{\mathcal{C}}(Y(\gamma, x))) \to \mathsf{Tm}_{\mathcal{C}}(A(\gamma, x)) \to \mathsf{Ty}_{\mathcal{C}},
\end{aligned}$$

along with a heterogeneous all-paths operation for the dependent type

$$AB \triangleq (x, y) \mapsto (a : A(\gamma, x)) \times B(\gamma, x, y, a)$$

over $(\gamma : \mathsf{y}(\Gamma))$.

We construct a homogeneous all-paths operation over $(\gamma : \mathsf{y}(\Gamma))$ for the type

$$(a : (x : X(\gamma)) \to A(\gamma, x)) \times (b : (x : X(\gamma)) \to (y : Y(\gamma, x)) \to B(\gamma, x, y, a(x))).$$

Construction. We write ab instead of (a, b), a_{lr} instead of (a_l, a_r), etc.

Our goal is to define the following:

$$\begin{aligned}\mathsf{hpath}_\Pi(\gamma, ab_l, ab_r) : (a_e : \forall x_{lre} \to [\![A]\!]_E((\mathsf{refl}_\Gamma(\gamma), x_e), a_l(x_l), a_r(x_r))) \\ \times (b_e : \forall x_{lre}\, y_{lre} \to [\![B]\!]_E((\mathsf{refl}_\Gamma(\gamma), x_e, y_e, a_e(x_e)), b_l(x_l, y_l), b_r(x_r, y_r))).\end{aligned}$$

Since Y is a family of contractible representable types, we can find paths in Y over any path in X.

$$\begin{aligned} y_0(\gamma, x) &\triangleq \mathsf{center}_Y(\gamma, x), \\ y_{0e}(\gamma, x_{lre}) :&\ [\![Y]\!]_E((\mathsf{refl}_\Gamma(\gamma), x_e), y_0(\gamma, x_l), y_0(\gamma, x_r)), \\ y_{0e}(\gamma, x_{lre}) &\triangleq [\![y_0]\!]_E(\mathsf{refl}_\Gamma, x_e). \end{aligned}$$

Now using the heterogeneous all-paths operation of AB, we obtain paths in A over any path in X.

$$\begin{aligned} a_e(\gamma, ab_{lr}, x_{lre}) \quad &: \ [\![A]\!]_E((\mathsf{refl}_\Gamma(\gamma), x_e), a_l(x_l), a_r(x_r)), \\ b_{0e}(\gamma, ab_{lr}, x_{lre}) \quad &: \ [\![B]\!]_E((\mathsf{refl}_\Gamma(\gamma), x_e, y_e, a_e(\gamma, a_{lr}, x_{lre})), b_l(x_l, y_0(\gamma, x_l)), b_r(x_r, y_0(\gamma, x_r))), \\ \langle a_e, b_{0e}\rangle(\gamma, ab_{lr}, x_{lre}) &\triangleq \mathsf{path}_{AB}(\gamma, x_e, y_{0e}(\gamma, x_{lre}), ab_l(x_l, y_0(\gamma, x_l)), ab_r(x_r, y_0(\gamma, x_r))). \end{aligned}$$

In order to define b_e, we transport b_{0e} over squares in Y. These squares are constructed using the fact that Y is contractible; since we already know that Id_- on representable types has a weakly stable elimination structure, we omit the precise construction of these squares.

$$s_Y(\gamma, x_{lre}, y_{lre}) : \mathsf{Id}_{\mathsf{Id}_Y}((\gamma, y_l, y_r), y_{0e}(\gamma, x_e), y_e).$$

We can now transport b_{0e} over s_Y.

$$\begin{aligned} I_B(\gamma, x_{lre}, a_{lre}, y_{lr}, b_{lr}, y_e) &\triangleq [\![B]\!]((\mathsf{refl}_\Gamma(\gamma), x_e, y_e, a_e), b_l, b_r), \\ T_{I_B}(\gamma, ab_{lr}, x_{lre}, y_{lre}) \quad &: \ I_B(\gamma, x_{lre}, a_e(\gamma, ab_{lr}, x_{lre}), b_l(xy_l)), b_r(xy_r), y_{0e}(\gamma, x_{lre})) \\ &\quad \simeq I_B(\gamma, x_{lre}, a_e(\gamma, ab_{lr}, x_{lre}), b_l(xy_l), b_r(xy_r), y_e), \\ T_{I_B}(x, y) &\triangleq \mathsf{DId}_{I_B}(\dots, s_Y(\gamma, x_{lre}, y_{lre})), \\ \overrightarrow{T_{I_B}}(\gamma, ab_{lr}, x_{lre}, y_{lre}) \quad &: \ \mathsf{isContr}((b_e : I_B(\gamma, ab_{lr}, x_{lre}, y_{lre})) \times \cdots), \\ \overrightarrow{T_{I_B}}(\gamma, ab_{lr}, x_{lre}, y_{lre}) &\triangleq T_{I_B}(\gamma, ab_{lr}, x_{lre}, y_{lre}).\overrightarrow{\mathsf{fun}}(b_{0e}(\gamma, ab_{lr}, x_{lre})), \\ b_e(\gamma, ab_{lr}, x_{lre}, y_{lre}) \quad &: \ [\![B]\!]((\mathsf{refl}_\Gamma(\gamma), x_e, y_e, a_e(\gamma, ab_{lr}, x_e)), b_l(xy_l), b_r(xy_r)), \\ b_e(\gamma, ab_{lr}, x_{lre}, y_{lre}) &\triangleq \overrightarrow{T_{I_B}}.\mathsf{center}(\gamma, ab_{lr}, x_e, y_e).1. \end{aligned}$$

Finally, we can pose

$$\mathsf{hpath}_\Pi(\gamma, ab_{lr}) \triangleq (\lambda x_{lre} \mapsto a_e(\gamma, a_{lr}, x_{lre}), \lambda x_{lre}\, y_{lre} \mapsto b_e(\gamma, ab_{lr}, x_{lre}, y_{lre})). \qquad \square$$

7. Proving external univalence

We fix a SOGAT $\mathcal{T}$ equipped with homotopy relations. Our goal is to prove that $\mathcal{T}$ satisfies external univalence, that is to equip the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ with weakly stable identity types satisfying function extensionality and saturation with respect to the homotopy relations of $\mathcal{T}$. In this section we essentially give a construction of this data (the weakly stable identity types) from the facts that every operation of $\mathcal{T}$ preserves the homotopy relations, and that the homotopy relations are equipped with some operations which essentially say that the homotopy relations should be reflexive and admit fillers of 1- and 2- dimensional cubes. More precisely, we rely on theorem 5.10 to construct the identity types, and the constructions of section 6 to satisfy the hypothesis of theorem 5.10, that is to construct reflexive equivalences and dependent equivalences with respect to some contractibility data, such that the contractible types have centers and all-paths operations.

7.1. **Internal model in reflexive equivalences.**

Assumption (A1). *We assume given a parametricity structure on $\mathcal{T}$, i.e. a section*

$$[\![-]\!] : \mathcal{T} \to \mathbf{PreReflGraph}(\mathcal{T})$$

of the projection map V. ⌟

By the universal property of $\mathcal{T}$, this amounts to equipping $\mathbf{PreReflGraph}(\mathcal{T})$ with an internal model of $\mathcal{T}$ that is displayed over $\mathcal{T}$.

Concretely, we have to interpret the sorts, operations and equations of $\mathcal{T}$ in $\mathbf{PreReflGraph}(\mathcal{T})$. The sorts have to be interpreted by Reedy types, which are typically given by the homotopy relations $(\sim)$, up to the fact that the Reedy types are more dependent than the homotopy relations. The representable sorts have to be interpreted by representable Reedy types.

Then to give an interpretation of an operation, we exactly need to show that it preserves the homotopy relations.

Depending on the theory $\mathcal{T}$, ensuring that the equations of $\mathcal{T}$ are satisfied in this model may be quite tricky. When $\mathcal{T}$ is a type theory with the usual β- and η- equalities, we have to use the relational definition of equivalences (example 4.3). While other definitions (such as half-adjoints equivalences, etc.) are equivalent up to homotopy, they do not seem to satisfy the necessary computational properties for this construction.

7.2. **Contractibility data.**

We define the following inductive families, internally to $\mathbf{Psh}(\mathcal{T})$.

$$\mathrm{isContr}^{\mathsf{rep},b}, \mathrm{isContr}^{b}, \mathrm{isContr}^{\mathsf{rep}}, \mathrm{isContr} : \mathsf{RepTy}_{\mathcal{T}} \to \mathcal{U}.$$

The families $\mathrm{isContr}^{\mathsf{rep},b}$ and $\mathrm{isContr}^{b}$ describe the *basic* contractible types. We introduce them so as to be able to state our last assumption later (assumption (A3)). The family $\mathrm{isContr}^{b}$ is generated by the following (non-recursive) constructors, for every generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$:

$$\begin{aligned}
\mathsf{contr}_{\mathbf{S},l} &: \forall (\sigma_l, \sigma_r : \partial\mathbf{S})\ (\sigma_e : [\![\partial\mathbf{S}]\!]_E(\sigma_l, \sigma_r))\ (a_l : \mathbf{S}(\sigma_l)) \\
&\qquad \to \mathrm{isContr}((a_r : \mathbf{S}(\sigma_r)) \times [\![\mathbf{S}]\!]_E(\sigma_e, a_l, a_r)), \\
\mathsf{contr}_{\mathbf{S},r} &: \forall (\sigma_l, \sigma_r : \partial\mathbf{S})\ (\sigma_e : [\![\partial\mathbf{S}]\!]_E(\sigma_l, \sigma_r))\ (a_r : \mathbf{S}(\sigma_r)) \\
&\qquad \to \mathrm{isContr}((a_l : \mathbf{S}(\sigma_l)) \times [\![\mathbf{S}]\!]_E(\sigma_e, a_l, a_r)), \\
\mathsf{contr}_{\mathbf{S},\sim} &: \forall (\sigma : \partial\mathbf{S})\ (x : \mathbf{S}(\sigma)) \\
&\qquad \to \mathrm{isContr}((y : \mathbf{S}(\sigma)) \times (x \sim_{\mathbf{S}(\sigma)} y)).
\end{aligned}$$

The constructor $\mathsf{contr}_{\mathbf{S},\sim}$ is included to make sure that we construct identity types satisfying saturation with respect to the homotopy relations $(\sim)$. The family $\mathrm{isContr}^{\mathsf{rep},b}$ is generated by the same constructors, but restricted to representable types. This means that the constructors $\mathsf{contr}_{\mathbf{S},l}$, $\mathsf{contr}_{\mathbf{S},r}$ and $\mathsf{contr}_{\mathbf{S},\sim}$ are only included for $\mathbf{S} : \mathsf{GenRepTy}_{\mathcal{T}}$. There is an evident map $\mathrm{isContr}^{\mathsf{rep},b}(A) \to \mathrm{isContr}^{b}(A)$ for $A : \mathsf{RepTy}_{\mathcal{T}}$.

The family $\mathrm{isContr}^{\mathsf{rep}}$ is inductively generated by the following constructors:

$$\begin{aligned}
&\mathsf{contr}^{\mathsf{rep}}_{b} : \forall A \to \mathrm{isContr}^{\mathsf{rep},b}(A) \to \mathrm{isContr}^{\mathsf{rep}}(A), \\
&\mathsf{contr}^{\mathsf{rep}}_{\mathbf{1}} : \mathrm{isContr}^{\mathsf{rep}}(\mathbf{1} \times \mathbf{1}), \\
&\mathsf{contr}^{\mathsf{rep}}_{\Sigma} : \forall (A : \mathsf{RepTy}_{\mathcal{T}})\ (B : \forall a \to \mathsf{RepTy}_{\mathcal{T}})\ (C : \forall a \to \mathsf{RepTy}_{\mathcal{T}})\ (D : \forall a\ b\ c \to \mathsf{RepTy}_{\mathcal{T}}) \\
&\qquad \to \mathrm{isContr}^{\mathsf{rep}}((a : A) \times B(a)) \\
&\qquad \to (\forall a\ b \to \mathrm{isContr}^{\mathsf{rep}}((c : C(a)) \times D(a,b,c))) \\
&\qquad \to \mathrm{isContr}^{\mathsf{rep}}(((a : A) \times (c : C(a))) \times ((b : B(a)) \times D(a,b,c))), \\
&\mathsf{contr}^{\mathsf{rep}}_{\cong} : \forall (A, B : \mathsf{RepTy}_{\mathcal{T}}) \to (\mathsf{Tm}_{\mathcal{T}}(A) \cong \mathsf{Tm}_{\mathcal{T}}(B)) \to \mathrm{isContr}^{\mathsf{rep}}(A) \to \mathrm{isContr}^{\mathsf{rep}}(B),
\end{aligned}$$

The family $\mathrm{isContr} : \mathsf{Ty}_{\mathcal{T}} \to \mathcal{U}$ is inductively generated by the following constructors:

$$\begin{aligned}
\mathsf{contr}_{\mathsf{rep}} &: \forall A \to \mathrm{isContr}^{\mathsf{rep}}(A) \to \mathrm{isContr}(A),\\
\mathsf{contr}_{b} &: \forall A \to \mathrm{isContr}^{b}(A) \to \mathrm{isContr}(A),\\
\mathsf{contr}_{\mathbf{1}} &: \mathrm{isContr}(\mathbf{1} \times \mathbf{1}),\\
\mathsf{contr}_{\Sigma} &: \forall (A : \mathsf{Ty}_{\mathcal{T}})\ (B : \forall a \to \mathsf{Ty}_{\mathcal{T}})\ (C : \forall a \to \mathsf{Ty}_{\mathcal{T}})\ (D : \forall a\ b\ c \to \mathsf{Ty}_{\mathcal{T}})\\
&\qquad \to \mathrm{isContr}((a : A) \times B(a))\\
&\qquad \to (\forall a\ b \to \mathrm{isContr}((c : C(a)) \times D(a,b,c)))\\
&\qquad \to \mathrm{isContr}(((a : A) \times (c : C(a))) \times ((b : B(a)) \times D(a,b,c))),\\
\mathsf{contr}_{\cong} &: \forall (A, B : \mathsf{Ty}_{\mathcal{T}}) \to (\mathsf{Tm}_{\mathcal{T}}(A) \cong \mathsf{Tm}_{\mathcal{T}}(B)) \to \mathrm{isContr}(A) \to \mathrm{isContr}(B),\\
\mathsf{contr}_{\Pi} &: \forall (X : \mathsf{RepTy}_{\mathcal{T}})\ (Y : X \to \mathsf{RepTy}_{\mathcal{T}})\\
&\qquad \to (\forall x \to \mathrm{isContr}^{\mathsf{rep}}(Y(x)))\\
&\qquad \to (A : X \to \mathsf{Ty}_{\mathcal{T}})\\
&\qquad \to (B : (x : X) \to (y : Y(x)) \to A(x) \to \mathsf{Ty}_{\mathcal{T}})\\
&\qquad \to (\forall x\ y \to \mathrm{isContr}((a : A(x)) \times B(x,y,a)))\\
&\qquad \to \mathrm{isContr}((a : (x : X) \to A(x))\\
&\qquad\qquad\qquad \times ((x : X) \to (y : Y(x)) \to B(x,y,a(x)))).
\end{aligned}$$

Now that we have defined this contractibility data, we observe that it is equipped, by definition, with the operations of lemma 6.10. Thus construction 6.11 provides the pre-reflexive equivalences model $\mathbf{PreReflEqv}(\mathcal{T})$ with respect to this contractibility data.

Lemma 7.1. *The section* $[\![-]\!] : \mathcal{T} \to \mathbf{PreReflGraph}(\mathcal{T})$ *factors through* $\mathbf{PreReflEqv}(\mathcal{T}) \to \mathbf{PreReflGraph}(\mathcal{T})$.

Proof. Note that $\mathbf{PreReflEqv}(\mathcal{T}) \to \mathbf{PreReflGraph}(\mathcal{T})$ is bijective on contexts and terms. Thus it suffices to consider the types.

We prove by induction on the types of $\mathcal{T}$ that for every type A (resp. representable type A), the Reedy type $[\![A]\!]$ (resp. representable Reedy type $[\![A]\!]$) is homotopical. For any generating type $\mathbf{S}$ (resp. generating representable type $\mathbf{S}$), this is handled by the constructors $\mathsf{contr}_{\mathbf{S},l}$ and $\mathsf{contr}_{\mathbf{S},r}$ (resp. $\mathsf{contr}^{\mathsf{rep}}_{\mathbf{S},l}$ and $\mathsf{contr}^{\mathsf{rep}}_{\mathbf{S},r}$). All of the other cases of the induction (for the $\mathbf{1}$-, Σ- and Π- type formers) follow from the fact that $\mathbf{PreReflEqv}(\mathcal{T})$ is a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF. □

7.3. **Reflexivity maps.** We now try to define a reflexivity structure (definition 6.15) over $\mathcal{T}$.

Definition 7.2 (Congruence operation)**.** Let $A :: \partial A \to \mathsf{RepTy}_{\mathcal{T}}$ and $B :: \partial B \to \mathsf{Ty}_{\mathcal{T}}$ be two global dependent types, where A is a representable dependent type.

A **congruence operation** from A to B consists of:

$$\begin{aligned}
\mathsf{ap}^{E}_{A,B} &:: \forall (\sigma : \partial A)\ (\sigma_e : [\![\partial A]\!]_E(\sigma,\sigma))\ (\sigma_r : [\![\partial A]\!]_R(\sigma,\sigma_e))\\
&\qquad (\tau : A(\sigma) \to \partial B)\\
&\qquad (\tau_e : \forall a_l\ a_r\ a_e \to [\![\partial B]\!]_E(\tau(a_l),\tau(a_r)))\\
&\qquad (\tau_r : \forall a\ a_e\ a_r \to [\![\partial B]\!]_R(\tau(a),\tau_e(a_e)))\\
&\qquad (b : (a : A(\sigma)) \to B(\tau(a)))\\
&\qquad\quad \to \forall a_l\ a_r\ (a_e : [\![A]\!]_E(\sigma_e, a_l, a_r)) \to [\![B]\!]_E(\tau_e(a_e), b(a_l), b(a_r)),\\
\mathsf{ap}^{R}_{A,B} &:: \forall \sigma\ \sigma_e\ \sigma_r\ \tau\ \tau_e\ \tau_r\ b\\
&\qquad\quad \to \forall a\ a_e\ a_r \to [\![B]\!]_R(\tau_r(a_r), \mathsf{ap}^{E}_{A,B}(\ldots, b, a_e)).
\end{aligned}$$
⌟

The representablility of A ensures that there is a type classifying the premises of a congruence operation, because $A(\sigma) \to \partial B$, $(a : A(\sigma)) \to B(\tau(a))$, etc., are types. Thus a congruence operation is fully determined by its evaluation at a suitable generic context.

Assumption (A2). *For every global dependent representable type A and generating type $\mathbf{S} : \mathsf{GenTy}_{\mathcal{T}}$, we have a congruence operation from $A :: \partial A \to \mathsf{RepTy}_{\mathcal{T}}$ to $\mathbf{S} :: \partial\mathbf{S} \to \mathsf{Ty}_{\mathcal{T}}$.* ⌟

In particular, the congruence operation from $\mathbf{1} :: \mathbf{1} \to \mathsf{RepTy}_{\mathcal{T}}$ to $\mathbf{S}$ is exactly a reflexivity operation for $\mathbf{S}$. When $\mathcal{T}$ is a first-order GAT, assumption (A2) reduces to the existence of reflexivity operations for every generating type $\mathbf{S}$.

Note that assumption (A2) is not ideal: it refers to an arbitrary representable type A, which can be seen as a telescope of basic representable types. Thus it may require data for every possible telescope shape. We would prefer to quantify over the generating representable types instead.

Lemma 7.3. *Every global dependent monomial type $A :: \partial A \to \mathsf{MonoTy}_{\mathcal{T}}$ can be equipped with a reflexivity operation.*

Proof. Since A is a monomial type, we can write

$$A(\sigma) = (\delta : \Delta(\sigma)) \to \mathbf{T}(\tau(\sigma,\delta))$$

for some $\Delta :: \partial A \to \mathsf{RepTy}_{\mathcal{T}}$ and $\tau :: \forall \sigma \to \Delta(\sigma) \to \partial\mathbf{T}$.

By definition of the section $[\![-]\!]$, we know that:

$$\begin{aligned}
[\![A]\!]_E(\sigma_l,\sigma_r,\sigma_e,f_l,f_r) &= \forall \delta_l\ \delta_r\ \delta_e \to [\![\mathbf{T}]\!]_E([\![\tau]\!]_E(\sigma_e,\delta_e), f_l(\delta_l), f_r(\delta_r)),\\
[\![A]\!]_R(\sigma,\sigma_e,\sigma_r,f,f_e) &= \forall \delta\ \delta_e\ \delta_r \to [\![\mathbf{T}]\!]_R([\![\tau]\!]_R(\sigma_r,\delta_r), f_e(\delta_e)).
\end{aligned}$$

We first pose:

$$\begin{aligned}
&\tau_e && : \forall \sigma\ \sigma_e\ \delta_l\ \delta_r\ \delta_e \to [\![\partial\mathbf{T}]\!]_E(\tau(\sigma_l,\delta_l)),\\
&\tau_e(\dots) && \triangleq [\![\tau]\!]_E((\sigma,\delta_l),(\sigma,\delta_r),(\sigma_e,\delta_e)),\\
&\tau_r && : \forall \sigma\ \sigma_e\ \sigma_r\ \delta\ \delta_e\ \delta_r \to [\![\partial\mathbf{T}]\!]_R(\tau(\sigma,\delta),\tau_e(\sigma_e,\delta_e)),\\
&\tau_r(\dots) && \triangleq [\![\tau]\!]_R((\sigma,\delta),(\sigma_e,\delta_e),(\sigma_r,\delta_r)).
\end{aligned}$$

By assumption (A2), we have a congruence operation $\mathsf{ap}_{\Delta,\mathbf{T}}$ from Δ to $\mathbf{T}$.

We can finally define the reflexivity operation for A:

$$\begin{aligned}
\mathsf{refl}^E_A(\sigma,\sigma_e,\sigma_r,f) &\triangleq \lambda \delta_l\ \delta_r\ \delta_e \mapsto \mathsf{ap}^E_{\Delta,\mathbf{T}}(\sigma_r,\tau(\sigma),\tau_e(\sigma_e),\tau_r(\sigma_r),f,\delta_e),\\
\mathsf{refl}^R_A(\sigma,\sigma_e,\sigma_r,f) &\triangleq \lambda \delta\ \delta_e\ \delta_r \mapsto \mathsf{ap}^R_{\Delta,\mathbf{T}}(\sigma_r,\tau(\sigma),\tau_e(\sigma_e),\tau_r(\sigma_r),f,\delta_r).
\end{aligned}$$ □

Lemma 7.4. *Every global dependent polynomial type $A :: \partial A \to \mathsf{PolyTy}_{\mathcal{T}}$ can be equipped with a reflexivity operation.*

Proof. We know that A is a telescope of monomial types and we can perform induction on its length. The case of the empty telescope is trivial.

If A is a non-empty telescope, we have

$$A(\sigma) = (b : B(\sigma)) \times (c : C(\sigma,b))$$

for some $B :: \partial A \to \mathsf{PolyTy}_{\mathcal{T}}$ and $C :: \forall \sigma \to B(\sigma) \to \mathsf{MonoTy}_{\mathcal{T}}$.

By the induction hypothesis, we have a reflexivity operation for B. By lemma 7.3, we have a reflexivity operation for C.

We can thus define:

$$\begin{aligned}
\mathsf{refl}^E_A(\sigma,\sigma_e,\sigma_r,(b,c)) &\triangleq (\mathsf{refl}^E_B(\sigma,\sigma_e,\sigma_r,b), \mathsf{refl}^E_C((\sigma,b),(\sigma_e,\mathsf{refl}^E_B(\dots,b)),(\sigma_r,\mathsf{refl}^R_B(\dots,b)),c)),\\
\mathsf{refl}^R_A(\sigma,\sigma_e,\sigma_r,(b,c)) &\triangleq (\mathsf{refl}^R_B(\sigma,\sigma_e,\sigma_r,b), \mathsf{refl}^R_C((\sigma,b),(\sigma_e,\mathsf{refl}^E_B(\dots,b)),(\sigma_r,\mathsf{refl}^R_B(\dots,b)),c)).
\end{aligned}$$ □

Lemma 7.5. *Every global dependent type $A :: \partial A \to \mathsf{Ty}_{\mathcal{T}}$ can be equipped with a reflexivity operation.*

Proof. By proposition 3.10, there is some polynomial type $A_0 :: \partial A \to \mathsf{PolyTy}_{\mathcal{T}}$ such that $\forall \sigma \to \mathsf{Tm}_{\mathcal{T}}(A_0(\sigma)) \cong \mathsf{Tm}_{\mathcal{T}}(A(\sigma))$. The results then follows from lemma 7.4. □

Thus, by proposition 6.16, we have a reflexivity structure over $\mathcal{T}$.

Now that we have a reflexivity structure, we obtain as in section 6.4 reflexive equivalences and dependent equivalences over $\mathcal{T}$.

7.4. **Centers and paths.** It remains to show that the contractible types have centers and all paths.

We assume that center and homogeneous all-paths operations are defined for the basic contractible types.

Assumption (A3). *For every constructor $c :: (\gamma : \Gamma) \to \mathrm{isContr}^b(A(\gamma))$ of $\mathrm{isContr}^b$, we have the following data:*

$$\begin{aligned}\mathsf{center}_c &:: \forall (\gamma : \Gamma) \to A(\gamma),\\ \mathsf{hpath}_c &:: \forall \gamma\ (\gamma_e : [\![\Gamma]\!]_E(\gamma,\gamma))\ (\gamma_r : [\![\Gamma]\!]_R(\gamma,\gamma_e))\\ &\qquad \to (x,y : A(\gamma)) \ \to [\![A]\!]_E(\gamma_e,x,y).\end{aligned}$$

Note that every constructor of $\mathrm{isContr}^b$ *is of this form.* ⌟

In assumption (A3) we have been careful not to mention refl_Γ; the assumption can be checked independently of the construction of the reflexivity maps, and independently of assumption (A2).

Lemma 7.6. *Every contractibility witness $c :: (\gamma : \Gamma) \to \mathrm{isContr}^b(A(\gamma))$ admits a center operation and a homogeneous all-paths operation.*

Proof. Fix a contractibility witness $c :: (\gamma : \Gamma) \to \mathrm{isContr}^b(A(\gamma))$. By definition of $\mathrm{isContr}^b$, there exists a constructor $d :: (\delta : \Delta) \to \mathrm{isContr}^b(B(\delta))$ of $\mathrm{isContr}^b$ such that $c = d[\delta]$ for some $\delta :: \Gamma \to \Delta$. In particular, $A = B[\delta]$.

We can now pose

$$\begin{aligned}c.\mathsf{center}(\gamma) &\triangleq \mathsf{center}_d(\delta(\gamma)),\\ c.\mathsf{hpath}(\gamma,x,y) &\triangleq \mathsf{hpath}_d(\delta(\gamma), [\![\delta]\!]_E(\mathsf{refl}^E_\Gamma(\gamma)), [\![\delta]\!]_R(\mathsf{refl}^R_\Gamma(\gamma)), x, y).\end{aligned}$$ □

We then proceed to extend this to arbitrary contractibility witnesses, relying on the constructions of section 6.4.

Lemma 7.7. *For every representable contractibility witness $c :: (\gamma : \Gamma) \to \mathrm{isContr}^{\mathsf{rep}}(A(\gamma))$, there is a center operation for c.*

Proof. By induction on c.

Constructor $\mathsf{contr}^{\mathsf{rep}}_b$**:**
By lemma 7.6.

Constructor $\mathsf{contr}^{\mathsf{rep}}_1$**:**
By construction 6.19.

Constructor $\mathsf{contr}^{\mathsf{rep}}_\Sigma$**:**
By construction 6.20. □

Lemma 7.8. *Let $c :: (\gamma : \Gamma) \to A(\gamma) \to \mathrm{isContr}^{\mathsf{rep}}(B(\gamma,a))$ be a global dependent family of contractibility witnesses. If we have a homogeneous all-paths operation for c over $\Gamma.A$, then we construct a heterogeneous all-paths operation for c.*

Proof. By construction 6.17, applied to the family restriction $\mathsf{RepTy}_\mathcal{T} \to \mathsf{Ty}_\mathcal{T}$. Checking the assumption relies on lemma 7.7. □

Lemma 7.9. *For every representable contractibility witness $c :: (\gamma : \Gamma) \to \mathrm{isContr}^{\mathsf{rep}}(A(\gamma))$, there is a homogeneous all-paths operation for c.*

Proof. By induction on c.

Constructor $\mathsf{contr}^{\mathsf{rep}}_b$**:**
By lemma 7.6.

Constructor $\mathsf{contr}^{\mathsf{rep}}_1$**:**
By construction 6.19.

Constructor $\mathsf{contr}^{\mathsf{rep}}_\Sigma$**:**
By construction 6.21 and lemma 7.8. □

We have now defined center and all-paths operations for all contractibility witnesses for representable types. We remark that this already equips the family of representable types with the structure of weakly stable identity types.

Proposition 7.10. *The family* $\mathsf{RepTy}_{\mathcal{T}}$ *is equipped with weakly stable identity types, where the introduction structure is given by the reflexive equivalences constructed in section 7.2.*

Proof. We use theorem 5.10. We have defined contractibility data, reflexive equivalences and dependent equivalences in section 7.2, and centers and paths of contractible types in lemma 7.7 and lemma 7.9. □

Lemma 7.11. *For every contractibility witness* $c :: \mathrm{isContr}(A)$, *there is a center operation for* c.

Proof. By induction on c.

Constructor contr_b**:**
By lemma 7.6.
Constructor $\mathsf{contr}_{\mathbf{1}}$**:**
By construction 6.19.
Constructor contr_{Σ}**:**
By construction 6.21.
Constructor contr_{Π}**:**
By construction 6.22. □

Lemma 7.12. *Let* $c :: (\gamma : \Gamma) \to (a : A(\gamma)) \to \mathrm{isContr}(B(\gamma, a))$ *be a global dependent family of contractibility witnesses. If we have a homogeneous all-paths operation for* c *over* $\Gamma.A$, *then we construct a heterogeneous all-paths operation for* c.

Proof. By construction 6.17, applied to the family restriction $\mathsf{Ty}_{\mathcal{T}} \to \mathsf{Ty}_{\mathcal{T}}$. Checking the assumption relies on lemma 7.11. □

Lemma 7.13. *For every contractibility witness* $c :: (\gamma : \Gamma) \to \mathrm{isContr}(A(\gamma))$, *there is a homogeneous all-paths operation for* c.

Proof. By induction on c.

Constructor contr_b**:**
By lemma 7.6.
Constructor $\mathsf{contr}_{\mathbf{1}}$**:**
By construction 6.19.
Constructor contr_{Σ}**:**
By construction 6.21 and lemma 7.12.
Constructor contr_{Π}**:**
By construction 6.23, proposition 7.10 and lemma 7.12. □

7.5. Main theorem.

We record the results of this section in the following theorem.

Theorem 7.14. *Let* $\mathcal{T}$ *be a SOGAT equipped with homotopy relations. If it satisfies assumption (A1), assumption (A2) and assumption (A3), then it satisfies external univalence, i.e. the* $(\Sigma, \Pi_{\mathsf{rep}})$*-CwF* $\mathcal{T}$ *can be equipped with weakly stable identity types satisfying function extensionality and saturation with respect to the homotopy relations.*

Proof. The weakly stable identity types are constructed using theorem 5.10. We have defined contractibility data, reflexive equivalences and dependent equivalences in section 7.2, and centers and paths of contractible types in lemma 7.11 and lemma 7.13. Saturation with respect to the homotopy relations follows from the constructors $\mathsf{contr}_{\mathsf{S},\sim}$ of $\mathrm{isContr}^{b}$. Function extensionality follows from the definition of the Π-types in $\mathbf{PreReflGraph}(\mathcal{C})$. □

7.6. **Necessary conditions.** The hypotheses of theorem 7.14 are not necessary conditions in the general setting. However, they become necessary for SOGATs without equations, and more generally for cofibrant SOGATs, i.e. for SOGATs that are retracts (in $\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}}}$) of SOGATs without equations.

Theorem 7.15. *Let $\mathcal{T}$ be a cofibrant SOGAT, i.e. a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF that is in the left class of the weak factorization system generated by $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}\}$.*

Assume that $\mathcal{T}$ is equipped with homotopy relations, such that for every generating representable sort $\mathbf{S} : \mathsf{GenRepTy}_{\mathcal{T}}$, the homotopy relation $(- \sim_{\mathbf{S}(-)} -)$ is a family of representable types.

If $\mathcal{T}$ satisfies external univalence, then it satisfies the conditions of theorem 7.14 (assumption (A1), assumption (A2) and assumption (A3)).

Proof. The identity types $(\simeq)$ of $\mathcal{T}$ induce contractibility data $\mathrm{isContr}_{\simeq}$ on $\mathcal{T}$, as constructed in construction 5.11.

The contractibility data $\mathrm{isContr}_{\simeq}$ satisfies the necessary closure conditions, so that the reflexive equivalences model $\mathbf{ReflEqv}(\mathcal{T})$ can be constructed by construction 6.13; we omit the details.

We consider the contextual core $\mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ of $\mathbf{ReflEqv}(\mathcal{T})$; its contexts can be obtained as iterated context extensions in $\mathbf{ReflEqv}(\mathcal{T})$. Since $\mathbf{ReflEqv}(\mathcal{T})$ has Σ-types, any context of $\mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ is isomorphic to a closed type of $\mathbf{ReflEqv}(\mathcal{T})$. In particular, any context of $\mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ satisfies the contractibility conditions of reflexive equivalences.

We now show that the projection morphism $V : \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T}) \to \mathcal{T}$ satisfies the right lifting property with respect to $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}\}$.

Case $I^{\mathsf{ty}} : \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type})$: Given an object $\Gamma : \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ and a type $A :: \mathbf{y}(\Gamma_V) \to \mathsf{Ty}_{\mathcal{T}}$, we have to extend A to a dependent reflexive equivalence (A, A_E, A_R) over Γ.

Since (Γ_E, Γ_R) satisfies the contractibility condition of a reflexive equivalence, it is equivalent to the reflexive equivalence Id_{Γ}. Up to this equivalence, we can then compute (A_E, A_R) as the dependent identity type $\mathsf{DId}_{\Gamma.A}$.

Case $I^{\mathsf{ty}_{\mathsf{rep}}} : \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}_{\mathsf{rep}})$: Similar to the case of I^{ty}. Given an object $\Gamma : \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ and a representable type $A :: \mathbf{y}(\Gamma_V) \to \mathsf{RepTy}_{\mathcal{T}}$, we have to extend A to a representable dependent reflexive equivalence (A, A_E, A_R) over Γ. We can compute (A_E, A_R) as the dependent identity type $\mathsf{DId}_{\Gamma.A}$. The assumption that the homotopy relation $(- \sim_{\mathbf{S}(-)} -)$ is a family of representable types whenever $\mathbf{S}$ is representable ensures that we can choose representable families for A_E and A_R.

Case $I^{\mathsf{tm}} : \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{A}\ \mathsf{type}) \to \mathsf{Free}_{\Sigma,\Pi_{\mathsf{rep}}}(\mathbf{\Gamma} \vdash \boldsymbol{a} : \boldsymbol{A})$: Given an object $\Gamma : \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$, a dependent reflexive equivalence A over Γ and a term $a :: (\gamma : \mathbf{y}(\Gamma_V)) \to \mathsf{Tm}_{\mathcal{T}}(A_V(\gamma))$, we have to show that a can be extended to a term of $\mathbf{ReflEqv}(\mathcal{T})$, that is we have to construct:

$$a_E : \forall \gamma_l\ \gamma_r\ (\gamma_e : \mathsf{Tm}_{\mathcal{T}}(\Gamma_E(\gamma_l, \gamma_r))) \to \mathsf{Tm}_{\mathcal{T}}(A_E(\gamma_e, a(\gamma_l), a(\gamma_e))),$$
$$a_R : \forall \gamma\ \gamma_e\ (\gamma_r : \mathsf{Tm}_{\mathcal{T}}(\Gamma_R(\gamma, \gamma_e))) \to \mathsf{Tm}_{\mathcal{T}}(A_R(\gamma_r, a(\gamma_l), a_E(\gamma_e))).$$

Up to the identification of (Γ_E, Γ_R) and (A_E, A_R) with respectively Id_{Γ} and $\mathsf{DId}_{\Gamma.A}$, a_E and a_R are just given by the dependent action of a on paths apd_a.

Note that we could not show the lifting property with respect to E^{tm}: there could be multiple ways to lift a same term from $\mathcal{T}$ to $\mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$, e.g. given by multiple definitions of the dependent action on paths.

Since $\mathcal{T}$ is $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}\}$-cellular, we obtain a section $\llbracket - \rrbracket : \mathcal{T} \to \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T})$ of $V : \mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T}) \to \mathcal{T}$. By composing this section with the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphism $\mathbf{ReflEqv}^{\mathsf{cxl}}(\mathcal{T}) \to \mathbf{PreReflGraph}(\mathcal{T})$, we equip $\mathbf{PreReflGraph}(\mathcal{T})$ with an internal model of $\mathcal{T}$, satisfying assumption (A1).

It remains to check assumption (A2) and assumption (A3).

The conditions of assumption (A2) are instances of the dependent action on paths.

Finally, assumption (A3) is proven by showing that induction on the inductive families $\mathrm{isContr}^{\mathsf{rep}}$ and $\mathrm{isContr}$ that for every $A :: \mathbf{y}(\Gamma) \to \mathsf{Ty}_{\mathcal{T}}$, we have

$$\forall \gamma \to \mathrm{isContr}(A(\gamma)) \to \mathrm{isContr}_{\simeq}(A(\gamma)),$$

since we already know that the types that are contractible with respect to $\mathsf{isContr}_{\simeq}$ have centers and all-paths. This amount to showing some closure conditions on $\mathsf{isContr}_{\simeq}$ that were already proven in the construction of $\mathbf{ReflEqv}(\mathcal{T})$. □

8. Applications

In this section, we apply theorem 7.14 to prove that some SOGATs satisfy external univalence.

8.1. Categories.

We first consider the first-order generalized algebraic theory $\mathcal{T}_{\mathbf{Cat}}$ of categories, equipped with the homotopy relations defined in example 4.2. We apply theorem 7.14 to show that it satisfies external univalence.

These constructions have been formalized in Agda [1], showing that for some concrete theory, assumption (A1), assumption (A2), assumption (A3) are syntactic enough as to be checked in a proof assistant.

We first need to check assumption (A1), that is to equip $\mathbf{PreReflGraph}(\mathcal{T}_{\mathbf{Cat}})$ with an internal model of $\mathcal{T}_{\mathbf{Cat}}$.

The sorts of objects, morphisms and equalities between morphisms are interpreted as follows:

$$\begin{aligned}
[\![\mathbf{ob}]\!]_E(\gamma_e) &\triangleq \lambda x_l\ x_r \mapsto x_l \cong x_r,\\
[\![\mathbf{hom}(x,y)]\!]_E(\gamma_e) &\triangleq \lambda f_l\ f_r \mapsto \mathbf{EqHom}([\![y]\!]_E(\gamma_e)\circ f_l, f_r \circ [\![x]\!]_E(\gamma_e)),\\
[\![\mathbf{eqhom}(f,g)]\!]_E(\gamma_e) &\triangleq \lambda p_l\ p_r \mapsto \mathbf{1},\\
[\![\mathbf{ob}]\!]_R(\gamma_r) &\triangleq \lambda x\ x_e \mapsto \mathbf{EqHom}(x_e, \mathbf{id}(x)),\\
[\![\mathbf{hom}(x,y)]\!]_R(\gamma_r) &\triangleq \lambda f\ f_e \mapsto \mathbf{1},\\
[\![\mathbf{eqhom}(x,y)]\!]_R(\gamma_r) &\triangleq \lambda p\ p_e \mapsto \mathbf{1}.
\end{aligned}$$

We then have to interpret the category operations, in a way that satisfies the category laws. We have to define the following components:

$$\begin{aligned}
[\![\mathbf{id}(x)]\!]_E(\gamma_e) &: \mathbf{EqHom}([\![x]\!]_E(\gamma_e)\circ \mathbf{id}(x(\gamma_l)), \mathbf{id}(x(\gamma_r))\circ [\![x]\!]_E(\gamma_e)),\\
[\![\mathbf{comp}(f,g)]\!]_E(\gamma_e) &: \mathbf{EqHom}([\![z]\!]_E(\gamma_e)\circ \mathbf{comp}(f(\gamma_l),g(\gamma_l)), \mathbf{comp}(f(\gamma_r),g(\gamma_r))\circ [\![x]\!]_E(\gamma_e)).
\end{aligned}$$

The components $[\![\mathbf{id}(x)]\!]_R$ and $[\![\mathbf{comp}(f,g)]\!]_R$ and are trivial, since $[\![\mathbf{hom}(-)]\!]_R(-)$ is the unit type.

The component $[\![\mathbf{id}(x)]\!]_E$ follows directly from the category laws.

For the remaining component $[\![\mathbf{comp}(f,g)]\!]_E$, remember that we have assumptions

$$\begin{aligned}
&[\![f]\!]_E(\gamma_e) : \mathbf{EqHom}([\![y]\!]_E(\gamma_e)\circ f(\gamma_l), f(\gamma_r)\circ [\![x]\!]_E(\gamma_e)), \text{and}\\
&[\![g]\!]_E(\gamma_e) : \mathbf{EqHom}([\![z]\!]_E(\gamma_e)\circ g(\gamma_l), g(\gamma_r)\circ [\![y]\!]_E(\gamma_e)).
\end{aligned}$$

We can then apply the category laws to derive $[\![\mathbf{comp}(f,g)]\!]_E(\gamma_e)$.

Because the components $-_E$ are propositional, the category laws are trivially satisfied in $\mathbf{PreReflGraph}(\mathcal{T}_{\mathbf{Cat}})$.

This finishes the definition of an internal model of $\mathcal{T}_{\mathbf{Cat}}$ in $\mathbf{PreReflGraph}(\mathcal{T}_{\mathbf{Cat}})$. We then have to check assumption (A2). Since $\mathcal{T}_{\mathbf{Cat}}$ is a first-order GAT, we just have to define reflexivity operations for every generating type. The reflexivity operation of objects is given by **id**, the reflexivity operations for morphisms is given by **irefl** and the reflexivity operation for equalities between morphisms is trivial.

It remains to check assumption (A3). We first unfold the definition of $\mathsf{isContr}^b$; it has the following constructors:

$$\begin{aligned}
\mathsf{contr}_{\mathbf{Ob},l} &: (x:\mathbf{Ob}) \to \mathsf{isContr}^b((y:\mathbf{Ob})\times(x\cong y)),\\
\mathsf{contr}_{\mathbf{Ob},r} &: (y:\mathbf{Ob}) \to \mathsf{isContr}^b((x:\mathbf{Ob})\times(x\cong y)),\\
\mathsf{contr}_{\mathbf{Ob},\sim} &: (x:\mathbf{Ob}) \to \mathsf{isContr}^b((y:\mathbf{Ob})\times(x\cong y)),\\
\mathsf{contr}_{\mathbf{Hom},l} &: \forall x_l\ x_r\ x_e\ y_l\ y_r\ y_e\ (f:\mathbf{Hom}(x_l,y_l))
\end{aligned}$$

[1]The Agda files are available at https://rafaelbocquet.gitlab.io/Agda/20221114_ExternalUnivalence/Cat.html.

$$
\begin{aligned}
&\quad \to \mathrm{isContr}^b((g : \mathbf{Hom}(x_r, y_r)) \times \mathbf{EqHom}(y_e \circ f \circ x_e^{-1}, g)),\\
\mathsf{contr}_{\mathbf{Hom},r} &: \forall x_l\ x_r\ x_e\ y_l\ y_r\ y_e\ (f : \mathbf{Hom}(x_r, y_r))\\
&\quad \to \mathrm{isContr}^b((g : \mathbf{Hom}(x_l, y_l)) \times \mathbf{EqHom}(y_e \circ g \circ x_e^{-1}, f)),\\
\mathsf{contr}_{\mathbf{Hom},\sim} &: \forall x\ y\ (f : \mathbf{Hom}(x, y))\\
&\quad \to \mathrm{isContr}^b((g : \mathbf{Hom}(x, y)) \times \mathbf{EqHom}(f, g)),\\
\mathsf{contr}_{\mathbf{EqHom},l} &: \forall x_l\ x_r\ x_e\ y_l\ y_r\ y_e\ f_l\ f_r\ f_e\ g_l\ g_r\ g_e\ (p : \mathbf{EqHom}(f_l, g_l))\\
&\quad \to \mathrm{isContr}^b(\mathbf{EqHom}(f_r, g_r) \times \mathbf{1}),\\
\mathsf{contr}_{\mathbf{EqHom},r} &: \forall x_l\ x_r\ x_e\ y_l\ y_r\ y_e\ f_l\ f_r\ f_e\ g_l\ g_r\ g_e\ (p : \mathbf{EqHom}(f_r, g_r))\\
&\quad \to \mathrm{isContr}^b(\mathbf{EqHom}(f_l, g_l) \times \mathbf{1}),\\
\mathsf{contr}_{\mathbf{EqHom},\sim} &: \forall x\ y\ f\ g\ (p : \mathbf{EqHom}(f, g))\\
&\quad \to \mathrm{isContr}^b(\mathbf{EqHom}(f, g) \times \mathbf{1}).
\end{aligned}
$$

However only $\mathsf{contr}_{\mathbf{Ob},l}$, $\mathsf{contr}_{\mathbf{Ob},r}$ and $\mathsf{contr}_{\mathbf{Ob},\sim}$ are interesting. In all of the other constructors, the type is isomorphic to $\mathbf{1}$, and verifying the conditions of assumption (A3) is then trivial. Furthermore, the constructors $\mathsf{contr}_{\mathbf{Ob},l}$, $\mathsf{contr}_{\mathbf{Ob},r}$ and $\mathsf{contr}_{\mathbf{Ob},\sim}$ are equivalent, and it suffices to consider $\mathsf{contr}_{\mathbf{Ob},l}$.

We have to construct the following terms:

$$
\begin{aligned}
\mathsf{center}_{\mathbf{Ob},l} &: \forall (x : \mathbf{Ob}) \to (y : \mathbf{Ob}) \times (f : x \cong y),\\
\mathsf{hpath}_{\mathbf{Ob},l} &: \forall x\ (x_e : x \cong x)\\
&\qquad (y_l : \mathbf{Ob})\ (f_l : x_l \cong y_l)\ (y_r : \mathbf{Ob})\ (f_r : x_r \cong y_r)\\
&\qquad \to (y_e : y_l \cong y_r)\\
&\qquad\quad \times \mathbf{EqHom}(y_e \circ f_l \circ x_e^{-1}, f_r)\\
&\qquad\quad \times \mathbf{EqHom}(x_e \circ f_l^{-1} \circ y_e^{-1}, f_r^{-1}).
\end{aligned}
$$

The input data for $\mathsf{hpath}_{\mathbf{Ob},l}$ can be described in the following diagram:

$$\begin{tikzcd}
x \arrow[r, "x_e", "\cong"'] \arrow[d, "f_l"', "\cong"] & x \arrow[d, "f_r", "\cong"'] \\
y_l & y_r .
\end{tikzcd}$$

Then $\mathsf{hpath}_{\mathbf{Ob},l}$ should be a filler of that open square.

They can be constructed as follows:

$$
\begin{aligned}
\mathsf{center}_{\mathbf{Ob},l}(x) &\triangleq (x, \mathbf{id}(x)),\\
\mathsf{hpath}_{\mathbf{Ob},l}(x, x_e, y_l, f_l, y_r, f_r) &\triangleq (f_r \circ x_e \circ f_l^{-1}, \mathbf{irefl}, \mathbf{irefl}).
\end{aligned}
$$

Thus we have checked assumption (A3). By theorem 7.14, the theory $\mathcal{T}_{\mathsf{Cat}}$ satisfies external univalence.

8.2. **Type theory with identity types.** We now show external univalence for the SOGAT $\mathcal{T}_{\mathsf{Id}}$ of a representable family equipped with weak identity types, equipped with the homotopy relations of example 4.3.

We first equip the inverse diagram model $\mathbf{PreReflGraph}(\mathcal{T}_{\mathsf{Id}})$ with an internal model of $\mathcal{T}_{\mathsf{Id}}$.

The sorts of types and terms are interpreted as follows:

$$
\begin{aligned}
[\![\mathbf{ity}]\!]_E(\gamma_e) &\triangleq \lambda A\ B \mapsto \mathrm{iEquiv}(A, B),\\
[\![\mathbf{itm}(A)]\!]_E(\gamma_e) &\triangleq \lambda x\ y \mapsto \mathbf{iTm}([\![A]\!]_E(\gamma_e, x, y)),\\
[\![\mathbf{ity}]\!]_R(\gamma_r) &\triangleq \lambda A\ E \mapsto \mathrm{isRefl}(E),\\
[\![\mathbf{itm}(A)]\!]_R(\gamma_r) &\triangleq \lambda x\ p \mapsto \mathbf{iTm}([\![A]\!]_R(\gamma_r, x, p)),
\end{aligned}
$$

where $\mathrm{iEquiv}(A,B)$ is the sort of relational equivalences between A and B, and

$$\begin{aligned}\mathrm{isRefl}(E) \triangleq{}& (P : \forall a \to \mathbf{iTm}(E(a,a)) \to \mathbf{iTy}) \\ & \times (\forall a \to \mathrm{isContr}((p : E(a,a)) \times P(p)))\end{aligned}$$

is the sort of reflexivity structures over an equivalence $E : \mathrm{iEquiv}(A,A)$.

We also have to interpret the operations **iId**, **irefl**, **iJ** and $\mathbf{iJ}_\beta$ in this model. For the operations **iId** and **irefl**, we have to define the following components:

$$\begin{aligned}
&[\![\mathbf{iId}(A,x,y)]\!]_E(\gamma_e) : \mathrm{iEquiv}(\mathbf{iId}(A(\gamma_l),x(\gamma_l),y(\gamma_l)),\mathbf{iId}(A(\gamma_r),x(\gamma_r),y(\gamma_r))),\\
&[\![\mathbf{iId}(A,x,y)]\!]_R(\gamma_r) : \mathrm{isRefl}([\![\mathbf{iId}(A,x,y)]\!]_E(\gamma_e)),\\
&[\![\mathbf{irefl}(A,x)]\!]_E(\gamma_e) \ \ : \mathbf{iTm}([\![\mathbf{iId}(A,x,x)]\!]_E(\gamma_e,\mathbf{irefl}(A(\gamma_l),x(\gamma_l)),\mathbf{irefl}(A(\gamma_r),x(\gamma_r)))),\\
&[\![\mathbf{irefl}(A,x)]\!]_R(\gamma_r) \ \ : \mathbf{iTm}([\![\mathbf{iId}(A,x,x)]\!]_R(\gamma_r,\mathbf{irefl}(A(\gamma),x(\gamma)),[\![\mathbf{irefl}(A,x)]\!]_E(\gamma_e))).
\end{aligned}$$

In other words, we have to show that **iId** preserves relational equivalences in a way that preserves reflexive equivalences, and that **irefl** preserves elements of these relations. We omit this standard proof.

This completes the definition of the internal model of $\mathcal{T}_{\mathsf{Id}}$ in $\mathbf{PreReflGraph}(\mathcal{T}_{\mathsf{Id}})$. We now have to check assumption (A2).

Let $A :: \partial A \to \mathsf{RepTy}_{\mathcal{T}_{\mathsf{Id}}}$ be a global dependent representable type. We have to construct congruence operations from A to **ity** and **itm**:

$$\begin{aligned}
\mathsf{ap}^E_{A,\mathbf{ity}} &:: \forall(\sigma : \partial A)(\sigma_e : [\![\partial A]\!]_E(\sigma,\sigma))(\sigma_r : [\![\partial A]\!]_R(\sigma,\sigma_e))\\
&\quad (B : A(\sigma) \to \mathbf{iTy})\\
&\quad \to \forall a_l\ a_r\ (a_e : [\![A]\!]_E(\sigma_e,a_l,a_r)) \to \mathrm{iEquiv}(B(a_l),B(a_r)),\\
\mathsf{ap}^R_{A,\mathbf{ity}} &:: \forall(\sigma : \partial A)(\sigma_e : [\![\partial A]\!]_E(\sigma,\sigma))(\sigma_r : [\![\partial A]\!]_R(\sigma,\sigma_e))\\
&\quad (B : A(\sigma) \to \mathbf{iTy})\\
&\quad \to \forall a\ a_e\ (a_r : [\![A]\!]_R(\sigma_r,a,a_e)) \to \mathrm{isRefl}(\mathsf{ap}^E_{A,\mathbf{ity}}(\sigma,\sigma_e,\sigma_r,B,a,a,a_e)),\\
\mathsf{ap}^E_{A,\mathbf{itm}} &:: \forall(\sigma : \partial A)(\sigma_e : [\![\partial A]\!]_E(\sigma,\sigma))(\sigma_r : [\![\partial A]\!]_R(\sigma,\sigma_e))\\
&\quad (B : A(\sigma) \to \mathbf{ity})\\
&\quad (B_e : \forall a_l\ a_r\ a_e \to \mathrm{iEquiv}(B(a_l),B(a_r)))\\
&\quad (B_r : \forall a\ a_e\ a_r \to \mathrm{isRefl}(B_e(a,a,a_e)))\\
&\quad (b : (a : A(\sigma)) \to \mathbf{iTm}(B(a)))\\
&\quad \to \forall a_l\ a_r\ (a_e : [\![A]\!]_E(\sigma_e,a_l,a_r)) \to B_e(a_l,a_r,a_e,b(a_l),b(a_r)),\\
\mathsf{ap}^R_{A,\mathbf{itm}} &:: \forall(\sigma : \partial A)(\sigma_e : [\![\partial A]\!]_E(\sigma,\sigma))(\sigma_r : [\![\partial A]\!]_R(\sigma,\sigma_e))\\
&\quad (B : A(\sigma) \to \mathbf{ity})\\
&\quad (B_e : \forall a_l\ a_r\ a_e \to \mathrm{iEquiv}(B(a_l),B(a_r)))\\
&\quad (B_r : \forall a\ a_e\ a_r \to \mathrm{isRefl}(B_e(a,a,a_e)))\\
&\quad (b : (a : A(\sigma)) \to \mathbf{iTm}(B(a)))\\
&\quad \to \forall a\ a_e\ (a_r : [\![A]\!]_R(\sigma_r,a,a_e)) \to B_r(a,a_e,a_r,b(a),\mathsf{ap}^E_{A,\mathbf{itm}}(\sigma,\sigma_e,\sigma_r,B,B_e,B_r,b,a,a,a_e)).
\end{aligned}$$

By proposition 3.10, we can assume without loss of generality that A is a telescope of basic representable sorts. Since the only representable sort of $\mathcal{T}_{\mathsf{Id}}$ is **itm**, this means that we have a dependent telescope $\Delta :: \partial A \to \mathbf{iTy}^\star$, with $A(\sigma) = \mathbf{iTm}^\star(\Delta(\sigma))$. We can then compute $[\![A]\!]_E$ and $[\![A]\!]_R$ and prove by induction on the telescope Δ that $(a_r : A(\sigma)) \times (a_e : [\![A]\!]_E(\sigma_e,a_l,a_r))$ and $(a_e : [\![A]\!]_E(\sigma_e,a,a)) \times (a_r : [\![A]\!]_R(\sigma_r,a,a_e))$ are contractible in $\mathbf{iTy}^\star$ (with respect to the inner identity types), which is sufficient to derive $\mathsf{ap}_{A,\mathbf{ity}}$ and $\mathsf{ap}_{A,\mathbf{itm}}$.

It remains to check assumption (A3). We first unfold the definition of $\mathrm{isContr}^b$ for this theory; it is given by the following constructors.

$$\begin{aligned}
\mathsf{contr}_{\mathbf{iTy},l} &: (A : \mathbf{iTy}) \to \mathrm{isContr}^b((B : \mathbf{iTy}) \times \mathrm{iEquiv}(A,B)) \\
\mathsf{contr}_{\mathbf{iTy},r} &: (B : \mathbf{iTy}) \to \mathrm{isContr}^b((A : \mathbf{iTy}) \times \mathrm{iEquiv}(A,B)) \\
\mathsf{contr}_{\mathbf{iTy},\sim} &: (A : \mathbf{iTy}) \to \mathrm{isContr}^b((B : \mathbf{iTy}) \times \mathrm{iEquiv}(A,B)) \\
\mathsf{contr}_{\mathbf{iTm},l} &: \forall A\ B\ (E : \mathrm{iEquiv}(A,B))\ (a : \mathbf{iTm}(A)) \\
&\qquad \to \mathrm{isContr}^b((b : \mathbf{iTm}(B)) \times \mathbf{iTm}(E(a,b))), \\
\mathsf{contr}_{\mathbf{iTm},r} &: \forall A\ B\ (E : \mathrm{iEquiv}(A,B))\ (b : \mathbf{iTm}(B)) \\
&\qquad \to \mathrm{isContr}^b((a : \mathbf{iTm}(A)) \times \mathbf{iTm}(E(a,b))), \\
\mathsf{contr}_{\mathbf{iTm},\sim} &: \forall A\ (x : \mathbf{iTm}(A)) \\
&\qquad \to \mathrm{isContr}^b((y : \mathbf{iTm}(A)) \times \mathbf{iTm}(\mathsf{Id}(A,x,y))).
\end{aligned}$$

We now check assumption (A3) for every constructor.

Constructor $\mathsf{contr}_{\mathbf{iTy},l}$:
We have to construct the following terms:

$$\begin{aligned}
\mathsf{center}_{\mathbf{iTy},l} &: \forall (A : \mathbf{iTy}) \to (B : \mathbf{iTy}) \times \mathrm{iEquiv}(A,B), \\
\mathsf{hpath}_{\mathbf{iTy},l} &: \forall A\ (A_e : \mathrm{iEquiv}(A,A)) \\
&\qquad (B_l : \mathbf{iTy})\ (E_l : \mathrm{iEquiv}(A,B_l))\ (B_r : \mathbf{iTy})\ (E_r : \mathrm{iEquiv}(A,B_r)) \\
&\qquad \to [\![A \mapsto (B : \mathbf{iTy}) \times (E : \mathrm{iEquiv}(A,B))]\!]_E(\ldots,(B_l,E_l),(B_r,E_r)).
\end{aligned}$$

The center is given by the identity equivalence on A.

$$\mathsf{center}_{\mathbf{iTy},l}(A) \triangleq (A, \mathsf{id}).$$

Defining hpath amounts to giving a composite and filler for the following open square of equivalences:

$$\begin{tikzcd}
A \arrow[r, "A_e", "\simeq"'] \arrow[d, "E_l"', "\simeq"] & A \arrow[d, "E_r", "\simeq"'] \\
B_l & B_r .
\end{tikzcd}$$

We omit this construction.

Constructor $\mathsf{contr}_{\mathbf{iTy},r}$:
Up to symmetry, this constructor is equivalent to $\mathsf{contr}_{\mathbf{iTy},l}$.

Constructor $\mathsf{contr}_{\mathbf{iTy},\sim}$:
This constructor is identical to $\mathsf{contr}_{\mathbf{iTy},l}$.

Constructor $\mathsf{contr}_{\mathbf{iTm},l}$:
We have to construct the following:

$$\begin{aligned}
\mathsf{center}_{\mathbf{iTm},l} &: \forall A\ B\ (E : \mathrm{iEquiv}(A,B))\ a \to (b : \mathbf{iTm}(B)) \times (p : \mathbf{iTm}(E(a,b))), \\
\mathsf{hpath}_{\mathbf{iTm},l} &: \forall (A : \mathbf{iTy})\ (A_e : \mathrm{iEquiv}(A,A))\ (A_r : \mathrm{isRefl}(A_e)) \\
&\qquad (B : \mathbf{iTy})\ (B_e : \mathrm{iEquiv}(B,B))\ (B_r : \mathrm{isRefl}(B_e)) \\
&\qquad (E : \mathrm{iEquiv}(A,B))\ E_e\ E_r \\
&\qquad (a : \mathbf{iTm}(A))\ (a_e : A_e(a,a))\ (a_r : A_r(a,a_e)) \\
&\qquad b_l\ (p_l : \mathbf{iTm}(E(a,b_l)))\ b_r\ (p_r : \mathbf{iTm}(E(a,b_r))) \\
&\qquad\ \to (b_e : \mathbf{iTm}(B_e(b_l,b_r))) \\
&\qquad\ \times (p_e : \mathbf{iTm}(E_e(a,a,a_e,b_l,b_r,b_e,p_l,p_r))).
\end{aligned}$$

The center is given by transporting a along the equivalence E:

$$\mathsf{center}_{\mathbf{iTm},l}(A,B,E,a) \triangleq E.\overrightarrow{\mathsf{fun}}(a).1.$$

Defining hpath amounts to giving a composite and filler for the following open dependent square

$$\begin{tikzcd} a \arrow[r, dash, "a_e"] \arrow[d, dash, "p_l"'] & a \arrow[d, dash, "p_r"] \\ b_l & b_r \end{tikzcd},$$

over the following square of equivalences

$$\begin{tikzcd} A \arrow[r, "A_e", "\simeq"'] \arrow[d, "E"', "\simeq"] \arrow[dr, phantom, "(E_e)"] & A \arrow[d, "E", "\simeq"'] \\ B \arrow[r, "\simeq", "B_e"'] & B \end{tikzcd}.$$

We also omit this construction.

Constructor $\mathsf{contr}_{\mathbf{iTm},r}$**:**
Up to symmetry, this constructor is equivalent to $\mathsf{contr}_{\mathbf{iTm},l}$.

Constructor $\mathsf{contr}_{\mathbf{iTm},\sim}$**:**
This is a version of $\mathsf{contr}_{\mathbf{iTm},l}$ that is specialized to the identity equivalence.

By theorem 7.14, the theory $\mathcal{T}_{\mathsf{Id}}$ satisfies external univalence.

8.3. **Other type formers.** If we extend $\mathcal{T}_{\mathsf{Id}}$ with additional type formers, the proofs of assumption (A2) and assumption (A3) remain the same. Only assumption (A1) needs to be checked; that is we have to give an interpretation of the additional operations in **PreReflGraph**$(\mathcal{T})$.

Giving an interpretation of type and term formers in **PreReflGraph**$(\mathcal{T})$ amounts to a more or less standard parametricity translation of the type and terms formers, similar to other parametricity translations found in the literature (Bernardy, Jansson, and Paterson 2012; Tabareau, Tanter, and Sozeau 2021). As in the case of identity types, we have to show that the type and term formers preserve equivalences and identification in a way that preserves identity equivalences and reflexivity identifications. Furthermore, these constructions have to be compatible with the various definitional equalities, such as the β- and η- rules for Σ- and Π- types.

For example, in the case of $\mathbf{1}$-, Σ- and Π- types, this translation mirrors at the inner level the outer level constructions of section 6:

$$\begin{aligned}
[\![\mathbf{1}]\!]_E(\gamma_e) &\triangleq \lambda t_l\ t_r \mapsto \mathbf{1}, \\
[\![\mathbf{1}]\!]_R(\gamma_r) &\triangleq \lambda t\ t_e \mapsto \mathbf{1}, \\
[\![\mathbf{\Sigma}(A,B)]\!]_E(\gamma_e) &\triangleq \lambda p_l\ p_r \mapsto (a_e : [\![A]\!]_E(\gamma_e, \pi_1(p_l), \pi_1(p_r))) \times (b_e : [\![B]\!]_E((\gamma_e,a_e), \pi_2(p_l), \pi_2(p_r))), \\
[\![\mathbf{\Sigma}(A,B)]\!]_R(\gamma_r) &\triangleq \lambda p\ p_e \mapsto (a_r : [\![A]\!]_R(\gamma_r, \pi_1(p), \pi_1(p_e))) \times (b_r : [\![B]\!]_R((\gamma_r,a_r), \pi_2(p), \pi_2(p_e))), \\
[\![\mathbf{\Pi}(A,B)]\!]_E(\gamma_e) &\triangleq \lambda f_l\ f_r \mapsto (\forall a_l\ a_r\ (a_e : [\![A]\!]_E(\gamma_e, a_l, a_r)) \to [\![B]\!]_E((\gamma_e,a_e), \mathbf{iapp}(f_l,a_l), \mathbf{iapp}(f_r,a_r)), \\
[\![\mathbf{\Pi}(A,B)]\!]_R(\gamma_r) &\triangleq \lambda f\ f_e \mapsto (\forall a\ a_e\ (a_r : [\![A]\!]_R(\gamma_r, a, a_e)) \to [\![B]\!]_R((\gamma_r,a_r), \mathbf{iapp}(f,a), f_e(a_e)).
\end{aligned}$$

An empty type $\mathbf{0}$ can be interpreted in the presence of a $\mathbf{1}$-type.

$$\begin{aligned}
[\![\mathbf{0}]\!]_E(\gamma_e) &\triangleq \lambda z_l\ z_r \mapsto \mathbf{1}, \\
[\![\mathbf{0}]\!]_R(\gamma_r) &\triangleq \lambda z\ z_e \mapsto \mathbf{1}.
\end{aligned}$$

Inductive types such as booleans or natural numbers can be interpreted provided that they support large elimination (or alternatively that sufficiently many indexed inductive types exist).

$$\begin{aligned}
[\![\mathbf{B}]\!]_E(\gamma_e)(\mathbf{true},\mathbf{true}) &\triangleq \mathbf{1},\\
[\![\mathbf{B}]\!]_E(\gamma_e)(\mathbf{true},\mathbf{false}) &\triangleq \mathbf{0},\\
[\![\mathbf{B}]\!]_E(\gamma_e)(\mathbf{false},\mathbf{true}) &\triangleq \mathbf{0},\\
[\![\mathbf{B}]\!]_E(\gamma_e)(\mathbf{false},\mathbf{false}) &\triangleq \mathbf{1}.
\end{aligned}$$

If $a :: \mathsf{Tm}_{\mathcal{T}}(A)$ is any axiom of the theory, that is a generating element of a closed sort of $\mathcal{T}$, then its interpretation in $\mathbf{PreReflGraph}(\mathcal{T})$ can be derived automatically from the reflexivity structure of $\mathcal{T}$:

$$\begin{aligned}
[\![a]\!]_E(\star) &\triangleq \mathsf{refl}^E_A(a),\\
[\![a]\!]_R(\star) &\triangleq \mathsf{refl}^R_A(a),
\end{aligned}$$

provided that refl^E_A and refl^R_A are well-defined (to be more precise we should consider extensions of the theory $\mathcal{T}$ by additional axioms).

The Uniqueness of Identity Proofs principle

$$\mathbf{uip} : \forall (A : \mathbf{iTy}) \to \mathrm{isSet}(A),$$

cannot be seen as an axiom in the previous sense, because $\forall(A : \mathbf{iTy}) \to \mathrm{isSet}(A)$ is not a sort. However it can still be interpreted in the pre-reflexive graphs model. It suffices to define

$$\begin{aligned}
[\![\mathbf{uip}(A)]\!]_E(\gamma_e) &: \mathrm{iEquiv}(\mathrm{isSet}(A(\gamma_l)), \mathrm{isSet}(A(\gamma_r))),\\
[\![\mathbf{uip}(A)]\!]_R(\gamma_r) &: \mathrm{isRefl}([\![\mathbf{uip}(A)]\!]_E(\gamma_e)).
\end{aligned}$$

When defining $[\![\mathbf{uip}(A)]\!]_E(\gamma_e)$, we have an equivalence $[\![A]\!]_E(\gamma_e)$ between $A(\gamma_l)$ and $A(\gamma_r)$, from which we can derive an equivalence between $\mathrm{isSet}(A(\gamma_l))$ and $\mathrm{isSet}(A(\gamma_r))$. Defining $[\![\mathbf{uip}(A)]\!]_R(\gamma_r)$ is straightforward using the fact that $\mathrm{isSet}(-)$ is a propositional type.

Theorem 8.1. *The SOGATs of type theories with weak identity types and any selection of type structures among:*

- *a* $\mathbf{1}$*-type;*
- Σ*-types;*
- Π*-types;*
- *an empty type* $\mathbf{0}$*, in the presence of* $\mathbf{1}$*;*
- *a boolean type with large elimination, in the presence of* $\mathbf{1}$ *and* $\mathbf{0}$*;*
- *a natural number type with large elimination, in the presence of* $\mathbf{1}$ *and* $\mathbf{0}$*;*
- *either weak or strict computation rules for any of the above type structures;*
- *any number of axioms, i.e. generating elements of closed representable sorts;*
- *the Uniqueness of Identity Proofs principle;*

all satisfy external univalence. □

8.4. **Type theories with universes.** Finally, we discuss the situation of universes. Tabareau et al. have achieved results that are similar to ours, but their work seemingly require the univalence axiom (Tabareau, Tanter, and Sozeau 2021, §6.4) We claim that by using universes à la Russell, they implicitly assume the existence of a coding function, that turns types into elements of the universe. By considering universes à la Tarski and without coding functions, there is no obstacle to the proof of external univalence, even in the presence of non-univalent universes.

We consider two SOGATs $\mathcal{T}_1$ and $\mathcal{T}_2$ that correspond to type theories with either a hierarchy of universes à la Tarski or a hierarchy of universes à la Coquand.

Definition 8.2. The SOGAT $\mathcal{T}_1$ is presented by the following signature

$$\begin{aligned}
\mathbf{ity} &: (i : \mathbb{N}) \to \mathsf{Ty},\\
\mathbf{iTy}_i &\triangleq \mathsf{Tm}(\mathbf{ity}_i),\\
\mathbf{itm} &: (i : \mathbb{N}) \to \mathbf{iTy} \to \mathsf{RepTy},
\end{aligned}$$

$$\begin{aligned}
&\mathbf{iTm}_i(A) \triangleq \mathsf{Tm}(\mathbf{itm}_i(A)),\\
&\mathbf{U} &&: (i : \mathbb{N}) \to \mathbf{iTy}_{i+1},\\
&\mathbf{El} &&: (i : \mathbb{N}) \to \mathbf{iTm}(\mathbf{U}_i) \to \mathbf{iTy}_i,\\
&\mathbf{u} &&: (i : \mathbb{N}) \to \mathbf{iTm}_{i+2}(\mathbf{U}_{i+1}),\\
&- &&: \mathbf{El}(\mathbf{u}_i) = \mathbf{U}_i,\\
&\mathbf{Lift} &&: (i : \mathbb{N}) \to \mathbf{iTy}_i \to \mathbf{iTy}_{i+1},\\
&\mathbf{lift} &&: (i : \mathbb{N})(A : \mathbf{iTy}_i) \to \mathbf{iTm}(A) \cong \mathbf{iTm}(\mathbf{Lift}(A)),\\
&\mathbf{iId} &&: (i : \mathbb{N})(A : \mathbf{iTy}_i)(x, y : \mathbf{iTm}(A)) \to \mathbf{iTy}_i,\\
&\mathbf{irefl} &&: (i : \mathbb{N})(A : \mathbf{iTy}_i)(x : \mathbf{iTm}(A)) \to \mathbf{iTm}_i(\mathbf{iId}(A, x, x)),\\
&\mathbf{iJ} &&: \ldots,\\
&\mathbf{iJ}_\beta &&: \ldots
\end{aligned}$$

The theory $\mathcal{T}_2$ is the extension of $\mathcal{T}_1$ with additional maps

$$\mathbf{c} : \mathbf{iTy}_i \to \mathbf{iTm}(\mathbf{U}_i)$$

that are inverses to the maps $\mathbf{El}$. ⌟

The two theories $\mathcal{T}_1$ and $\mathcal{T}_2$ may seem equivalent, since they their syntaxes can be identified.

Proposition 8.3. *The map $\mathbf{0}_{\mathcal{T}_1} \to \mathbf{0}_{\mathcal{T}_2}$ is an isomorphism (of models of $\mathcal{T}_1$).*

Proof. It suffices to show that the maps $\mathbf{El}$ have inverses in $\mathbf{0}_{\mathcal{T}_1}$, which follows from normalization for the type theory $\mathcal{T}_1$. □

However, while $\mathbf{0}_{\mathcal{T}_1} \to \mathbf{0}_{\mathcal{T}_2}$ is an isomorphism, this is not the case for the map $\mathcal{T}_1 \to \mathcal{T}_2$ between their coclassifying $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs. We will show that $\mathcal{T}_1$ satisfies external univalence while $\mathcal{T}_2$ does not.

Definition 8.4. We define homotopy relations on both $\mathcal{T}_1$ and $\mathcal{T}_2$, analogously to the homotopy relations defined in example 4.3.

$$\begin{aligned}
A \sim_{\mathbf{ity}_i} B \quad &\triangleq \mathrm{iEquiv}(A, B),\\
x \sim_{\mathbf{itm}_i(A)} y &\triangleq \mathbf{iTm}(\mathbf{iId}(A, x, y)).
\end{aligned}$$

Lemma 8.5. *The theory $\mathcal{T}_1$ satisfies external univalence with respect to the homotopy relations defined in definition 8.4.*

Proof. We use theorem 7.14. Assumption (A2) and assumption (A3) can be checked in the same way as in section 8.2; we omit the proof.

We equip $\mathbf{PreReflGraph}(\mathcal{T}_1)$ with the structure of an internal model of $\mathcal{T}_1$.

The sorts of types and terms are interpreted as follows:

$$\begin{aligned}
[\![\mathbf{ity}_i]\!]_E(\gamma_e) \quad &\triangleq \lambda A\ B \mapsto \mathrm{iEquiv}_i(A, B),\\
[\![\mathbf{itm}_i(A)]\!]_E(\gamma_e) &\triangleq \lambda x\ y \mapsto \mathbf{iTm}_i([\![A]\!]_E(\gamma_e, x, y)),\\
[\![\mathbf{ity}_i]\!]_R(\gamma_r) \quad &\triangleq \lambda A\ E \mapsto \mathrm{isRefl}_i(E),\\
[\![\mathbf{itm}_i(A)]\!]_R(\gamma_r) &\triangleq \lambda x\ p \mapsto \mathbf{iTm}_i([\![A]\!]_R(\gamma_r, x, p)),
\end{aligned}$$

where $\mathrm{iEquiv}_i(A, B)$ is the sort of relational equivalences between A and B in Ty_i, and

$$\begin{aligned}
\mathrm{isRefl}(E) \triangleq\ &(P : \forall a \to \mathbf{iTm}_i(E(a, a)) \to \mathbf{iTy}_i)\\
&\times (\forall a \to \mathrm{isContr}((p : E(a, a)) \times P(p)))
\end{aligned}$$

is the sort of reflexivity structures in Ty_i over an equivalence $E : \mathrm{iEquiv}_i(A, A)$.

The universes are interpreted as follows; note that the relation between elements of the universe is not equivalence, but identification of the codes:

$$[\![\mathbf{U}_i]\!]_E(\gamma_e) \quad : \ \mathrm{iEquiv}_{i+1}(\mathbf{U}_i, \mathbf{U}_i),$$

$$\begin{aligned}
&[\![\mathbf{U}_i]\!]_E(\gamma_e) && \triangleq \lambda A\ B \mapsto \mathbf{iId}_{\mathbf{U}_i}(A,B),\\
&[\![\mathbf{U}_i]\!]_R(\gamma_r) && :\ \mathrm{isRefl}_{i+1}([\![\mathbf{U}_i]\!]_E(\gamma_e)),\\
&[\![\mathbf{U}_i]\!]_R(\gamma_r) && \triangleq \lambda A\ e \to \mathbf{iId}(e, \mathbf{irefl}_{\mathbf{U}_i}(A)),\\
&[\![\mathbf{El}_i(A)]\!]_E(\gamma_e) && :\ \mathrm{iEquiv}_i(\mathbf{El}(A(\gamma_l)), \mathbf{El}(A(\gamma_r))),\\
&[\![\mathbf{El}_i(A)]\!]_E(\gamma_e) && \triangleq \mathsf{idtoeqv}([\![A]\!]_E(\gamma_e)),\\
&[\![\mathbf{El}_i(A)]\!]_R(\gamma_r) && :\ \mathrm{isRefl}(\mathsf{idtoeqv}([\![A]\!]_E(\gamma_e))),\\
&[\![\mathbf{El}_i(A)]\!]_R(\gamma_r) && \triangleq (\text{follows from } [\![A]\!]_R(\gamma_r) : \mathbf{iId}([\![A]\!]_E(\gamma_e), \mathbf{irefl}(A(\gamma)))).
\end{aligned}$$

Interpreting the lifting operations is straightforward, and the identity types can be interpreted as in section 8.2.

This completes the construction of the internal model of $\mathcal{T}_1$ in $\mathbf{PreReflGraph}(\mathcal{T}_1)$. By theorem 7.14, the theory $\mathcal{T}_1$ satisfies external univalence. □

Lemma 8.6. *The theory $\mathcal{T}_2$ does not satisfies external univalence with respect to the homotopy relations defined in definition 8.4.*

Proof. Assuming that $\mathcal{T}_2$ satisfies external univalence, the action of $\mathbf{c}$ on paths determines a map

$$(A, B : \mathbf{iTy}_0) \to \mathrm{iEquiv}(A, B) \to \mathbf{iTm}(\mathbf{iId}(\mathcal{U}_0, \mathbf{c}(A), \mathbf{c}(B)))$$

in the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}_2$.

This map can be interpreted into any model of $\mathcal{T}_2$; in particular it can be interpreted in the standard model **Set**. Since we can assume without loss of generality that there exists two sets A and B that are equivalent but not equal, $\mathcal{T}_2$ cannot satisfy external univalence. (In a set-theoretic metatheory, we can choose $A = \{1\}$ and $B = \{2\}$. In a type-theoretic metatheory, we can add new redundant codes to the universes of **Set**, e.g. we define $\mathcal{U}'_0 \triangleq \mathcal{U}_0 + \{A, B\}$ with $\mathsf{El}(A) = \mathsf{El}(B) = \mathbf{1}$.) □

Theorem 8.7. *The SOGATs of type theories with a $\mathbb{N}$-indexed hierarchy of universes à la Tarski, weak identity types and any selection of type structures among:*

- $\mathbf{1}$*-types;*
- Σ*-types;*
- Π*-types;*
- *empty types* $\mathbf{0}$*, in the presence of* $\mathbf{1}$*;*
- *boolean types, in the presence of* $\mathbf{1}$ *and* $\mathbf{0}$*;*
- *natural number types, in the presence of* $\mathbf{1}$ *and* $\mathbf{0}$*;*
- *either weak or strict computation rules for the above type structures;*
- *any number of axioms, i.e. postulated elements of closed representable sorts types, such as the univalence axiom;*
- *the Uniqueness of Identity Proofs principle;*

all satisfy external univalence. □

9. Future work

9.1. **Semantic study of external univalence.** In this paper, we have defined external univalence as a property of the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$. The advantage of this approach is that we did not have to consider the semantics of $\mathcal{T}$ at all.

In future work, we plan to study how external univalence for a SOGAT $\mathcal{T}$ is related to properties of the category $\mathbf{Mod}_{\mathcal{T}}$ of models of $\mathcal{T}$. In particular, we plan to show claim 4.8, which says that $\mathcal{T}$ satisfies external univalence exactly when the category $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}}$ of contextual models of $\mathcal{T}$, equipped with suitable classes of maps, is a left semi-model category. We also plan to show that the notion of Morita equivalence between type theories, which was introduced by Isaev (2018), can be captured at the level of the $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs: given a morphism $\mathcal{T}_1 \to \mathcal{T}_2$ of SOGATs that satisfy external univalence, the adjunction between the left semi-model categories $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}_1}$ and $\mathbf{Mod}^{\mathsf{cxl}}_{\mathcal{T}_2}$ is a Quillen adjunction if and only if $\mathcal{T}_1 \to \mathcal{T}_2$ preserves the identity types, and a Quillen equivalence if and only if $\mathcal{T}_1 \to \mathcal{T}_2$ is additionally a weak equivalence in $\mathbf{CwF}_{\Sigma,\Pi_{\mathsf{rep}},\mathsf{Id}_{ws}}$.

9.2. **Strictification.** It is rather inconvenient that our methods only equip $\mathcal{T}$ with weakly stable identity types. It does not seem possible to equip $\mathcal{T}$ with strictly stable identity types in general. Instead, we may want to strictify the identity types, i.e. faithfully embed $\mathcal{T}$ into a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF with strictly stable identity types.

Conjecture 9.1. Let $\mathcal{T}$ be a SOGAT equipped with homotopy relations. If $\mathcal{T}$ satisfies external univalence, then there exists a $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id})$-CwF $\mathcal{C}$ with strictly stable identity types and a $(\Sigma, \Pi_{\mathsf{rep}})$-CwF morphism $\mathcal{T} \to \mathcal{C}$ that weakly preserves identity types and is essentially surjective on types and terms. ⌟

The known strictification methods cannot be applied to this situation. For example, the local universes method (Lumsdaine and Warren 2015) requires more Π-types than available in $\mathcal{T}$.

It is however possible to strictify the identity types in the special case of first-order generalized algebraic theories without equations.

Theorem 9.2. *Let $\mathcal{T}$ be a first-order generalized algebraic theory without equations, i.e. an $\{I^{\mathsf{ty}}, I^{\mathsf{tm}}\}$-cellular Σ-CwF.*

If $\mathcal{T}$ satisfies external univalence with respect to a choice of homotopy relations, then $\mathcal{T}$ can be equipped with strictly stable identity types satisfying saturation with respect to the homotopy relations.

Proof. This is proven for $\{I^{\mathsf{ty}}, I^{\mathsf{tm}}\}$-cellular CwFs (without Σ) in [Bocquet 2022, Theorems 1 and 2], but the methods can be generalized to $\{I^{\mathsf{ty}}, I^{\mathsf{tm}}\}$-cellular Σ-CwFs. □

9.3. **Embedding theories into richer models.** While we have shown that it is possible to transport structures over homotopies for any SOGAT that satisfies external univalence, this only holds for structures that are expressible in the language of the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$. This language is not sufficiently expressive for all applications.

It would be desirable to conservatively embed $\mathcal{T}$ into a richer language that allows for the specification of additional structures and properties. A good candidate for this richer language is (any variant of) Homotopy Type Theory.

Conjecture 9.3 (Weak embedding into HoTT)**.** Let $\mathcal{T}$ be a SOGAT equipped with homotopy relations satisfying external univalence. Then there exists a model $\mathcal{C}$ of (some variant of) HoTT equipped with an univalent internal model of $\mathcal{T}$ such that the induced $(\Sigma, \Pi_{\mathsf{rep}})$-morphism $\mathcal{T} \to \mathcal{C}$ is essentially surjective on terms. ⌟

Conjecture 9.4 (Strict embedding into HoTT)**.** Let $\mathcal{T}$ be a SOGAT equipped with homotopy relations satisfying external univalence. Then there exists a model $\mathcal{C}$ of (some variant of) HoTT equipped with an univalent internal model of $\mathcal{T}$ such that the induced $(\Sigma, \Pi_{\mathsf{rep}})$-morphism $\mathcal{T} \to \mathcal{C}$ is bijective on terms. ⌟

These conjectures should be seen as ∞-categorical variants of the following 1-categorical theorem:

Theorem 9.5. *Let $\mathcal{T}$ be any SOGAT. Then there exists a model $\mathcal{C}$ of extensional type theory equipped with an internal model of $\mathcal{T}$ such that the induced $(\Sigma, \Pi_{\mathsf{rep}})$-morphism $\mathcal{T} \to \mathcal{C}$ is bijective on terms.*

Proof sketch. We define $\mathcal{C}$ as the presheaf topos $\widehat{\mathcal{T}}$. Then the Yoneda embedding $\mathcal{y} : \mathcal{T} \to \widehat{\mathcal{T}}$ is a pseudo-morphism of $(\Sigma, \Pi_{\mathsf{rep}})$-CwFs, and bijective on terms. Relying on the fact that $\mathcal{T}$ is $\{I^{\mathsf{ty}}, I^{\mathsf{ty}_{\mathsf{rep}}}, I^{\mathsf{tm}}, E^{\mathsf{tm}}\}$-cellular, we can construct a strict replacement $y : \mathcal{T} \to \widehat{\mathcal{T}}$ of the Yoneda embedding, along with a 2-cell $\mathcal{y} \cong y$. This strict replacement is also bijective on terms. □

An ∞-categorical version of this argument gives intuition for why conjecture 9.3 should hold. Indeed, when a SOGAT $\mathcal{T}$ satisfies external univalence, the $(\Sigma, \Pi_{\mathsf{rep}})$-CwF $\mathcal{T}$ is a $(\Sigma, \Pi_{\mathsf{rep}}, \mathsf{Id}_{ws})$-CwF, which should correspond to some ∞-category with representable maps. By the ∞-categorical Yoneda lemma, we can faithfully embed this ∞-category into an ∞-topos of ∞-categorical presheaves. We can finally interpret HoTT into this ∞-topos. Unfortunately, turning this this informal proof idea into a proper proof is not straightforward.

When studying the computational properties of type theories, such as canonicity and normalization properties, it is typical to rely on the interpretation of extensional type theory into some presheaf categories. A solution of these conjectures would provide a good setting for the study of some homotopical properties of type theories, such as homotopy canonicity and normalization up to homotopy.

References

Adamek, J. and J. Rosicky (1994). *Locally Presentable and Accessible Categories*. London Mathematical Society Lecture Note Series. Cambridge University Press. DOI: 10.1017/CBO9780511600579.

Ahrens, Benedikt, Krzysztof Kapulkin, and Michael Shulman (2015). "Univalent categories and the Rezk completion". In: *Mathematical Structures in Computer Science* 25.5, pp. 1010–1039. DOI: 10.1017/S0960129514000486.

Ahrens, Benedikt, Paige Randall North, et al. (2020). "A Higher Structure Identity Principle". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '20. Saarbrücken, Germany: Association for Computing Machinery, pp. 53–66. ISBN: 9781450371049. DOI: 10.1145/3373718.3394755. URL: https://doi.org/10.1145/3373718.3394755.

Altenkirch, Thorsten and Ambrus Kaposi (2015). "Towards a Cubical Type Theory without an Interval". In: *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*. Ed. by Tarmo Uustalu. Vol. 69. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 3:1–3:27. DOI: 10.4230/LIPIcs.TYPES.2015.3. URL: https://doi.org/10.4230/LIPIcs.TYPES.2015.3.

Altenkirch, Thorsten, Ambrus Kaposi, and Michael Shulman (2022). "Towards Higher Observational Type Theory". In: *28th International Conference on Types for Proofs and Programs, TYPES 2022*. Ed. by Pierre-Marie Pédrot. LS2N, University of Nantes. URL: https://types22.inria.fr/files/2022/06/TYPES_2022_paper_37.pdf.

Angiuli, Carlo et al. (2021). "Syntax and models of Cartesian cubical type theory". In: *Mathematical Structures in Computer Science* 31.4, pp. 424–468. DOI: 10.1017/S0960129521000347.

Atkey, Robert, Neil Ghani, and Patricia Johann (2014). "A relationally parametric model of dependent type theory". In: *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*. Ed. by Suresh Jagannathan and Peter Sewell. ACM, pp. 503–516. DOI: 10.1145/2535838.2535852. URL: https://doi.org/10.1145/2535838.2535852.

Awodey, Steve (2018). "Natural models of homotopy type theory". In: *Mathematical Structures in Computer Science* 28.2, pp. 241–286. DOI: 10.1017/S0960129516000268.

Bernardy, Jean-Philippe, Patrik Jansson, and Ross Paterson (Mar. 2012). "Proofs for Free: Parametricity for Dependent Types". In: *J. Funct. Program.* 22.2, pp. 107–152. ISSN: 0956-7968. DOI: 10.1017/S0956796812000056. URL: https://doi.org/10.1017/S0956796812000056.

Blanc, Georges (1978). "Équivalence Naturelle Et Formules Logiques En Théorie des Catégories". In: *Archive for Mathematical Logic* 19.1, pp. 131–137. DOI: 10.1007/bf02011874.

Bocquet, Rafaël (2020). "Coherence of strict equalities in dependent type theories". In: *CoRR* abs/2010.14166. arXiv: 2010.14166. URL: https://arxiv.org/abs/2010.14166.

– (2022). "Strictification of Weakly Stable Type-Theoretic Structures Using Generic Contexts". In: *27th International Conference on Types for Proofs and Programs (TYPES 2021)*. Ed. by Henning Basold, Jesper Cockx, and Silvia Ghilezan. Vol. 239. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 3:1–3:23. ISBN: 978-3-95977-254-9. DOI: 10.4230/LIPIcs.TYPES.2021.3. URL: https://drops.dagstuhl.de/opus/volltexte/2022/16772.

Brunerie, Guillaume (2016). "On the homotopy groups of spheres in homotopy type theory". In: *CoRR* abs/1606.05916. arXiv: 1606.05916. URL: http://arxiv.org/abs/1606.05916.

Castellan, Simon, Pierre Clairambault, and Peter Dybjer (2019). "Categories with Families: Unityped, Simply Typed, and Dependently Typed". In: *CoRR* abs/1904.00827. arXiv: 1904.00827. URL: http://arxiv.org/abs/1904.00827.

Clairambault, Pierre and Peter Dybjer (2014). "The biequivalence of locally cartesian closed categories and Martin-Löf type theories". In: *Math. Struct. Comput. Sci.* 24.6. DOI: 10.1017/S0960129513000881. URL: https://doi.org/10.1017/S0960129513000881.

Coquand, Thierry (2013). "Presheaf model of type theory". Available at https://www.cse.chalmers.se/~coquand/presheaf.pdf.

– (2019). "Canonicity and normalization for dependent type theory". In: *Theor. Comput. Sci.* 777, pp. 184–191. DOI: 10.1016/j.tcs.2019.01.015. URL: https://doi.org/10.1016/j.tcs.2019.01.015.

Dybjer, Peter (1995). "Internal Type Theory". In: *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers*. Ed. by Stefano Berardi and Mario Coppo. Vol. 1158. Lecture Notes in Computer Science. Springer, pp. 120–134. DOI: 10.1007/3-540-61780-9_66. URL: https://doi.org/10.1007/3-540-61780-9%5C_66.

Freyd, Peter J. (1976). "Properties Invariant within Equivalence Types of Categories". In.

Henry, Simon (2020). *The language of a model category*. www.uwo.ca/math/faculty/kapulkin/seminars/hottestfiles/Henry-2020-01-23-HoTTEST.pdf.

Hofmann, Martin (1999). "Semantical Analysis of Higher-Order Abstract Syntax". In: *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*. IEEE Computer Society, pp. 204–213. DOI: 10.1109/LICS.1999.782616. URL: https://doi.org/10.1109/LICS.1999.782616.

Isaev, Valery (2017). "Model structures on categories of models of type theories". In: *Mathematical Structures in Computer Science* 28, pp. 1695–1722.

– (2018). "Morita equivalences between algebraic dependent type theories". In: *CoRR* abs/1804.05045. arXiv: 1804.05045. URL: http://arxiv.org/abs/1804.05045.

Kaposi, Ambrus, András Kovács, and Thorsten Altenkirch (2019). "Constructing quotient inductive-inductive types". In: *Proc. ACM Program. Lang.* 3.POPL, 2:1–2:24. DOI: 10.1145/3290315. URL: https://doi.org/10.1145/3290315.

Kapulkin, Krzysztof and Peter LeFanu Lumsdaine (2018). "The homotopy theory of type theories". In: *Advances in Mathematics* 337, pp. 1–38. ISSN: 0001-8708. DOI: https://doi.org/10.1016/j.aim.2018.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0001870818303062.

– (Apr. 2021). "Homotopical inverse diagrams in categories with attributes". In: *Journal of Pure and Applied Algebra* 225, p. 106563. DOI: 10.1016/j.jpaa.2020.106563.

Kapulkin, Krzysztof and Karol Szumiło (Sept. 2017). "Internal Language of Finitely Complete $(\infty, 1)$-categories". In: *Selecta Mathematica* 25. DOI: 10.1007/s00029-019-0480-0.

Kraus, Nicolai and Christian Sattler (2017). "Space-Valued Diagrams, Type-Theoretically (Extended Abstract)". In: *CoRR* abs/1704.04543. arXiv: 1704.04543. URL: http://arxiv.org/abs/1704.04543.

Lumsdaine, Peter LeFanu and Michael A. Warren (2015). "The Local Universes Model: An Overlooked Coherence Construction for Dependent Type Theories". In: *ACM Trans. Comput. Log.* 16.3, 23:1–23:31. DOI: 10.1145/2754931. URL: https://doi.org/10.1145/2754931.

Makkai, Michael (1995). "First Order Logic with Dependent Sorts, with Applications to Category Theory". Available at http://www.math.mcgill.ca/makkai/folds/.

– (1998). "Towards a categorical foundation of mathematics". In: *Logic Colloquium'95: Proceedings of the Annual European Summer Meeting of the Association of Symbolic Logic, held in Haifa, Israel, August 9-18, 1995*. Vol. 11. Citeseer, pp. 153–191.

Nguyen, Hoang Kim and Taichi Uemura (2022). *∞-type theories*. DOI: 10.48550/ARXIV.2205.00798. URL: https://arxiv.org/abs/2205.00798.

Reynolds, John C. (1983). "Types, Abstraction and Parametric Polymorphism". In: *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France, September 19-23, 1983*. Ed. by R. E. A. Mason. North-Holland/IFIP, pp. 513–523.

Riehl, Emily (2008). *Factorization Systems*. Available at http://www.math.jhu.edu/~eriehl/factorization.pdf.

Ringer, Talia et al. (Sept. 2019). "Ornaments for proof reuse in Coq". English (US). In: *10th International Conference on Interactive Theorem Proving, ITP 2019*. Ed. by John Harrison, John O'Leary, and Andrew Tolmach. Leibniz International Proceedings in Informatics, LIPIcs. Germany: Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing. DOI: 10.4230/LIPIcs.ITP.2019.26.

Shulman, Michael (2015). "Univalence for inverse diagrams and homotopy canonicity". In: *Mathematical Structures in Computer Science* 25.5, pp. 1203–1277. DOI: 10.1017/S0960129514000565.

Shulman, Michael (2017). “Brouwer’s fixed-point theorem in real-cohesive homotopy type theory”. In: *Mathematical Structures in Computer Science* 28, pp. 856–941.

– (2022). *Towards a Third-Generation HOTT*. https://www.cmu.edu/dietrich/philosophy/hott/slides/shulman-2022-04-28.pdf.

Streicher, Thomas (2014). “Semantics of Type Theory Formulated in Terms of Representability”. Available at https://www2.mathematik.tu-darmstadt.de/~streicher/FIBR/natmod.pdf.

Tabareau, Nicolas, Éric Tanter, and Matthieu Sozeau (2021). “The Marriage of Univalence and Parametricity”. In: *J. ACM* 68.1, 5:1–5:44. DOI: 10.1145/3429979. URL: https://doi.org/10.1145/3429979.

Uemura, Taichi (2019). “A General Framework for the Semantics of Type Theory”. In: *CoRR* abs/1904.04097. arXiv: 1904.04097. URL: http://arxiv.org/abs/1904.04097.

– (2021). “Abstract and concrete type theories”. PhD thesis. Institute for Logic, Language and Computation. URL: https://dare.uva.nl/search?identifier=41ff0b60-64d4-4003-8182-c244a9afab3b.

Univalent Foundations Program, The (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book.

Email address: bocquet@inf.elte.hu

DEPARTMENT OF PROGRAMMING LANGUAGES AND COMPILERS, EÖTVÖS LORÁND UNIVERSITY, BUDAPEST, HUNGARY