SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Type theoretic semantics for first order logic

av

Oskar Berndal

2020 - No M1

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Type theoretic semantics for first order logic

Oskar Berndal

Självständigt arbete i matematik 30 högskolepoäng, avancerad nivå

Handledare: Peter Lumsdaine

2020

Abstract

Whereas the semantics of first order logic are well-understood, many questions remain regarding the semantics of type theory. There is not even an established and unified notion of what precisely is a type theory.

In a recent work by Uemura, a general notion of type theories is proposed together with semantics for these type theories. The aim of this work is to present a type theory within this framework such that its semantics recovers the semantics for first order logic.

The main obstacle is the mismatch between what one takes as a morphism in the semantics: In first order logic one takes the functional relations whereas in type theory one essentially takes its terms. In order to bridge this gap we introduce terms for definite descriptions to the type theory.

Acknowledgements

I would like to thank Peter for making me feel that my word is taken seriously. I would like to thank Axel for opening his house and listening to me a Friday evening when I really needed it. I would like to thank Tiger for his insistent purring and demands of scratches.

Contents

1 Background **3**
1.1 Primer on first order logic 5
1.1.1 Specification of first order logic 5
1.1.2 The syntactic category 7
1.1.3 Functorial semantics of first order logic 8
1.2 Basics of type theory 10

2 Syntax **13**
2.1 First order logic as a type theory 13
2.1.1 Raw syntax 14
2.1.2 Rules for symbols from the signature 15
2.1.3 Logical and structural rules 16
2.1.4 Rules for axiom terms 18
2.2 Some syntactic conveniences 19
2.2.1 Heyting structure on term model 22
2.3 Translating judgements into definite description free fragment . . 23
2.3.1 Specification and soundness clauses 24
2.3.2 Definitions and partial soundness results 26
2.3.3 Substitution and soundness 38
2.3.4 Translating sequents 44
2.4 The syntactic equivalence 45

3 Semantics **48**
3.1 Modelling type theory 48
3.1.1 Logical structure on judgement fibrations 49
3.1.2 Signature and axiom structures 55
3.1.3 A comment on the framework 56
3.1.4 Logical structures on the standard pair 58
3.2 Recovering functorial semantics for first order logic 63

4 Further directions **65**
4.1 Semantics for a proof relevant version with homotopy type theory 65

1 Background

Let us broadly state some of the central concepts of mathematical logic.

Formal system
: Some set of syntactic expressions, together with some rules for derivability.

Model (of a formal system)
: A mathematical structure which in particular supports an interpretation of the syntactic expressions.

Semantics
: A specification of precisely what things we shall consider to be models.

Soundness
: All derivable expressions of the formal system are true when interpreted in a model.

For instance, one may take a formal system of vector spaces and one may give a semantics where the models of this system are $\mathbb{R}^n$ for all $n \geq 0$. Soundness would tell us that all things which we can formally derive are true when interpreted in $\mathbb{R}^n$. For a taste of how this works, let the following be a derivable expression of our formal system,

$$\forall\, x, y : X \;\; x + y = (y + x) + \mathbf{0}$$

which reasonably would, when interpreted in $\mathbb{R}^3$, correspond to the fact that

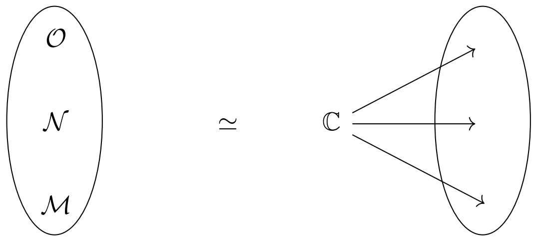
for any tuples $(a_1, b_1, c_1), (a_2, b_2, c_2) \in \mathbb{R}^3$*, we have that*

$$(a_1, b_1, c_1) + (a_2, b_2, c_2) = [(a_2, b_2, c_2) + (a_1, b_1, c_1)] + (0, 0, 0).$$

The families formal systems that we will be interested in are theories in first order logic and type theories.

In a common semantics for first order logic the models are sets-with-structure and the logical formulas correspond to subsets (think Venn diagrams). This would be more general than the semantics provided above because there are many finite dimensional vector spaces which are not $\mathbb{R}^n$ (although every finite dimensional vector space is isomorphic to some $\mathbb{R}^n$).

There is a generalization of sets-with-structure-semantics where instead of sets one takes objects of some category, instead of structure one takes arrows in this category, and instead of subsets one takes subobjects in the category. One can prove that this semantics has an initial category $\mathbb{C}$, which classifies all other models. This means that for any theory of first order logic, there is a category $\mathbb{C}$ such that for any (sufficiently nice) category $\mathcal{C}$ there is a correspondence between models in $\mathcal{C}$ and (sufficiently nice) functors $\mathbb{C} \to \mathcal{C}$,



whence one says that theories of first order logic have an *initial object*.

The story is messier for type theories. It is harder to define a general notion for what a type theory is. The purpose of this work is to introduce a type theory such that we can recover the functorial semantics of first order logic within the semantics of the type theory. To achieve this end, we use a framework introduced by Uemura in [7] to equip every type theory (for some such notion) with a semantics that is sound and that comes with its own initial object $\mathcal{I}$.

The way that this will be done is that we first show that the initial objects $\mathbb{C}$ and $\mathcal{I}$ are equivalent as categories. Then we show that certain models in the type theoretic sense are the same things as sufficiently nice functors from $\mathcal{I}$ into sufficiently nice categories.

The main obstruction in trying to give a type theory such that $\mathbb{C}$ and $\mathcal{I}$ are equivalent is in a mismatch between the way that the arrows of the respective categories are formed: In $\mathbb{I}$, the arrows are immediately formed from terms in the type theory. In $\mathbb{C}$ they are however formed from those propositions which one can prove are functional relations. So on the first order logic side, the arrows do not "immediately appear" in the syntax, whereas on the type theory side, the have to "immediately appear" in the syntax.

The way we rectify this is that, aside from giving constructors corresponding to the logical connectives and logical rules, we add terms corresponding to definite descriptions to the type theory. These are in essence terms corresponding to functional relations.

Because we can have proof-terms as arguments in type theory one does not run into the same kinds of problems that one does if one tries to add definite descriptions to first order logic. The only issue is that as they do not have a counterpart in the syntax of first order logic, one somehow needs to eliminate them in order to translate back and forth.

Section 1 recounts syntax and semantics of first order logic, and gives a short introduction to type theory.

Section 2 introduces and investigates the syntax of our type theory with the main goal of proving that $\mathcal{I}$ and $\mathbb{C}$ are equivalent.

Section 3 introduces the semantics of our type theory with the main goal of showing how the semantics of first order logic correspond to a subset of the type theoretic semantics.

Section 4 concludes with some further directions for the semantics of the type theory.

1.1 Primer on first order logic

Our formulation of first order logic is based on the one presented by Jacobs in Chapter 4 of [3]. The expressions of first order logic consist mainly of sequents of the form

$$x : \sigma.\, y : \sigma.\, z : \mathbb{N} \mid x =_\sigma f(y),\, \varphi(z) \vdash \psi(y, z)$$

where the sequent is supposed to be read as saying that

Let x, y be arbitrary σ and z be an arbitrary natural number.
If x and $f(y)$ are equal, and z satisfies the proposition φ,
then y, z satisfy the proposition ψ.

Left of the vertical line is a specification of what variables can appear in the expressions to the right and $x : \sigma$ reflects that x only can take σ-values for some such notion.

1.1.1 Specification of first order logic

A sequent system is a specification of the set of sequents together with some rules for making inferences. These rules are of the form

$$\frac{A \qquad B \qquad C}{D}$$

and mean that if A, B and C are derivable sequents then so is D. The sequents of first order logic are all of the form

$$\Gamma \mid \mathcal{J}$$

where Γ will be called the sort-context and $\mathcal{J}$ the judgement. The judgements come in three forms,

sort judgement	sort term	proposition judgement	proposition sequent
$\sigma\ \texttt{sort}$	$t : \sigma$	$\varphi\ \texttt{prop}$	$\Theta \vdash \vartheta$

where the Θ of the proposition sequent is a list of propositions. Let us first define what our propositions are. The notion for capturing this is that of a *signature*. A first-order signature Σ is specified by

- the set of sorts corresponding to what kind of values the terms are allowed to take,
- non-logical symbols and what kind of arguments they accept.

We have

$$\sigma \in \texttt{sorts}\,\Sigma \qquad \Xi \in \texttt{atoms}_n\Sigma, \quad \texttt{arity}\,\Xi = \vec{\sigma} \in (\texttt{sorts}\,\Sigma)^n$$

$$f \in \texttt{functions}_n\Sigma, \quad \texttt{arity}\, f = (\sigma_1, \ldots, \sigma_n, \sigma) \in (\texttt{sorts}\,\Sigma)^{n+1}$$

with rules for these symbols,

$$\frac{}{\sigma\ \texttt{sort}} \qquad \frac{\Gamma \mid t_1 : \sigma_1 \qquad \cdots \qquad \Gamma \mid t_n : \sigma_n}{\Gamma \mid f(t_1,\, \ldots,\, t_n) : \sigma}$$

$$\frac{\Gamma \mid t_1 : \sigma_1 \qquad \cdots \qquad \Gamma \mid t_n : \sigma_n}{\Gamma \mid \Xi(t_1, \ldots, t_n)\ \texttt{prop}}$$

Next let us introduce the structural rules, which concern how our variables and terms interact

$$\frac{\Gamma.\, \Delta \mid \mathcal{J} \qquad \sigma\ \texttt{sort}}{\Gamma.\, x : \sigma.\, \Delta \mid \mathcal{J}} \qquad \frac{\Gamma.\, x : \sigma.\, \Delta \mid \mathcal{J} \qquad \Gamma \mid t : \sigma}{\Gamma.\, \Delta \mid \mathcal{J}\,[x \backslash t\,]} \qquad \frac{\sigma\ \texttt{sort}}{x : \sigma \mid x : \sigma}$$

Basic rules for forming compound formulas

$$\frac{\Gamma \mid \varphi\ \texttt{prop} \qquad \Gamma \mid \psi\ \texttt{prop}}{\Gamma \mid \varphi \bigcirc \psi\ \texttt{prop}} \qquad \frac{\Gamma.\, x : \sigma \mid \varphi\ \texttt{prop}}{\Gamma \mid Q\ x : \sigma\ \varphi\ \texttt{prop}} \qquad \frac{\Gamma \mid \varphi\ \texttt{prop}}{\Gamma \mid \varphi \vdash \varphi}$$

Here $\bigcirc$ stands for the binary connectives $\wedge, \vee, \Rightarrow$ and Q stands for the quantifiers $\forall, \exists$. Finally let us look at the rules corresponding to logical laws. Let first introduce the abbreviation

$$\Gamma \mid \Theta\ \texttt{prop} \quad \text{for} \quad \Gamma \mid \theta_1\ \texttt{prop}\ \ldots\ \Gamma \mid \theta_n\ \texttt{prop}$$

for $\Theta = \theta_1, \ldots, \theta_n$.

$$\frac{\Gamma \mid \Theta\ \texttt{prop}}{\Gamma \mid \Theta \vdash \top} \qquad \frac{\Gamma \mid \Theta \vdash \bot \qquad \Gamma \mid \vartheta\ \texttt{prop}}{\Gamma \mid \Theta \vdash \vartheta} \qquad \frac{\Gamma \mid t : \sigma \qquad \Gamma \mid \tau : \sigma}{\Gamma \mid t =_\sigma \tau\ \texttt{prop}}$$

$$\frac{\Gamma \mid t : \sigma \qquad \Gamma \mid \Theta\ \texttt{prop}}{\Gamma \mid \Theta \vdash t =_\sigma t}$$

$$\frac{\Gamma.\, x : \sigma \mid \varphi\ \texttt{prop} \qquad \Gamma \mid \Theta \vdash t =_\sigma \tau \qquad \Gamma \mid \Theta \vdash \varphi\,[x \backslash t\,]}{\Gamma \mid \Theta \vdash \varphi\,[x \backslash \tau\,]}$$

$$\frac{\Gamma \mid \Theta \vdash \varphi \qquad \Gamma \mid \Theta \vdash \psi}{\Gamma \mid \Theta \vdash \varphi \wedge \psi} \qquad \frac{\Gamma \mid \Theta \vdash \varphi \wedge \psi}{\Gamma \mid \Theta \vdash \varphi} \qquad \frac{\Gamma \mid \Theta \vdash \varphi \wedge \psi}{\Gamma \mid \Theta \vdash \psi}$$

$$\frac{\Gamma \mid \Theta \vdash \varphi \qquad \Gamma \mid \psi\ \texttt{prop}}{\Gamma \mid \Theta \vdash \varphi \vee \psi} \qquad \frac{\Gamma \mid \Theta \vdash \psi \qquad \Gamma \mid \varphi\ \texttt{prop}}{\Gamma \mid \Theta \vdash \varphi \vee \psi}$$

$$\frac{\Gamma \mid \Theta \vdash \varphi \vee \psi \qquad \Gamma \mid \Theta,\, \varphi \vdash \vartheta \qquad \Gamma \mid \Theta,\, \psi \vdash \vartheta}{\Gamma \mid \Theta \vdash \vartheta}$$

$$\frac{\Gamma \mid \Theta, \varphi \vdash \psi}{\Gamma \mid \Theta \vdash \varphi \Rightarrow \psi} \qquad \frac{\Gamma \mid \Theta \vdash \varphi \Rightarrow \psi \qquad \Gamma \mid \Theta \vdash \varphi}{\Gamma \mid \Theta \vdash \psi}$$

$$\frac{\Gamma \mid t : \sigma \qquad \Gamma .\, x : \sigma \mid \varphi \ \texttt{prop} \qquad \Gamma \mid \Theta \vdash \varphi\,[x \backslash t]}{\Gamma \mid \Theta \vdash \exists x : \sigma\ \varphi}$$

$$\frac{\Gamma \mid \Theta \vdash \exists x : \sigma\ \varphi \qquad \Gamma \mid \vartheta \ \texttt{prop} \qquad \Gamma .\, x : \sigma \mid \Theta, \varphi \vdash \vartheta}{\Gamma \mid \Theta \vdash \vartheta}$$

$$\frac{\Gamma .\, x : \sigma \mid \Theta \vdash \varphi \qquad \Gamma \mid \Theta \ \texttt{prop}}{\Gamma \mid \Theta \vdash \forall x : \sigma\ \varphi} \qquad \frac{\Gamma \mid \Theta \vdash \forall x : \sigma\ \varphi \qquad \Gamma \mid t : \sigma}{\Gamma \mid \Theta \vdash \varphi\,[x \backslash t]}$$

Next we introduce the rules for adding axioms in order to obtain a *theory*. Given a signature Σ, for any sequent $\Gamma \mid \Theta \vdash \vartheta$ such that

$$\Gamma \mid \theta_i \ \texttt{prop} \qquad\qquad \Gamma \mid \vartheta \ \texttt{prop}$$

are all derivable (where θ_i are the components of Θ) will be called a *good sequent*. A theory $\mathcal{T}$ over Σ is a set of good sequents (called axioms). The derivable sequents given by Σ and $\mathcal{T}$ is the extension of the one given by Σ by the following for each axiom of $\mathcal{T}$,

$$\frac{}{\ \mid \Theta \vdash \vartheta}$$

where the empty context signifies that all the variables of the sequent must be bound.

1.1.2 The syntactic category

Definition 1.1. The *syntactic category* $\mathbb{C}_{\Sigma,\mathcal{T}}$ of a first order theory $\mathcal{T}$ in signature Σ is a category whose

- objects are sort-contexts Γ with a list of propositions $\Theta = \theta_i$ such that for each i

  $$\Gamma \mid \theta_i \ \texttt{prop}$$

  is derivable, such an object will be denoted by $\{\Gamma \mid \Theta\}$

- morphisms from $\{\Gamma \mid \Theta\}$ to $\{\Delta \mid \Lambda\}$ are propositions ϕ with $\Gamma .\, \Delta \mid \phi \ \texttt{prop}$ and

  $$\Gamma .\, \Delta \mid \phi \vdash \theta_i, \quad \Gamma .\, \Delta \mid \phi \vdash \lambda_j \qquad \Gamma .\, \vec{x} : \Delta .\, \vec{y} : \Delta \mid \phi\,[\vec{x}],\, \phi\,[\vec{y}] \vdash \vec{x} =_{\vec{\sigma}} \vec{y}$$

  $$\Gamma \mid \Theta \vdash \exists \Delta\ \phi$$

  modulo provable equivalence, $\phi \sim \psi$ if both

  $$\Gamma .\, \Delta \mid \phi \vdash \psi \quad \text{and} \quad \Gamma .\, \Delta \mid \psi \vdash \phi$$

  are derivable from $\mathcal{T}$.

1.1.3 Functorial semantics of first order logic

A commonly used semanticcs for first order logic is the Heyting categories that will be defined in this section. The propositions of a first-order theory will be interpreted as subobjects and $\Gamma \mid \varphi \vdash \psi$ being derivable will correspond to ψ being interpreted as a smaller subobject of Γ than φ.

Definition 1.2. A *cartesian category* is a finitely complete category.

A cartesian category is suitable for interpreting conjunction, via subobject intersections, and equality, via equalizers. We can also interpret substitution via pullbacks of subobjects, which will be denoted by $f^*(U)$ for a subobject U. Next we will show how to interpret existential quantification.

Definition 1.3. Given a subobject $I \subseteq \Gamma'$ and an arrow $f : \Gamma' \to \Gamma$ assume that there is a subobject $\Sigma_f I \subseteq \Gamma$ such that for any subobject $U \subseteq \Gamma$

$$I \leq f^* U \qquad \Leftrightarrow \qquad \Sigma_f I \leq U \ .$$

We then call $\Sigma_f I$ the *dependent sum* of I along f.

Remark 1.4. Dependent sums are uniquely defined from their input data, should they exist.

Definition 1.5. A *regular category* is a finitely complete category such that all subobjects have dependent sums along all morphisms.

Lemma 1.6. Let a pullback square $f' \circ g' = f \circ g$ in a regular category be given. Then we have

$$\Sigma_{g'} g^*(X) = f'^* \Sigma_f(X)$$

for any subobject X.

Next up are the models for binary disjunctions and false, which allow us to interpret finite disjunctions.

Definition 1.7. Given two subobjects $I, J \subseteq \Gamma$, assume that there is a subobject $I \cup J$ such that for any subobject $U \subseteq \Gamma$

$$I \cup J \leq U \qquad \Leftrightarrow \qquad I \leq U \ \text{ and } \ J \leq U \ .$$

We then call $I \cup J \subseteq \Gamma$ the *union* of I and J.

Definition 1.8. Suppose we have a subobject $\bot_\Gamma \subseteq \Gamma$ such that for any other subobject $U \subseteq \Gamma$ we have $\bot \leq U$. We then call $\bot_\Gamma$ the *initial subobject* of Γ.

Remark 1.9. Unions and initial subobjects are uniquely defined from their input data, should they exist.

Definition 1.10. A cartesian category is said to have *finite well-behaved unions* if it has all binary unions and initial subobjects, and they commute with pullbacks. More precisely, for any $f : \Gamma' \to \Gamma$ and $I, J \subseteq \Gamma$ we have

- $f^*(\bot_\Gamma) = \bot_{\Gamma'}$
- $f^*(I \cup J) = f^*(I) \cup f^*(J)$

Finally the models for universal quantification.

Definition 1.11. Given a subobject $I \subseteq \Gamma'$ and an arrow $f : \Gamma' \to \Gamma$ assume that there is a subobject $\Pi_f I \subseteq \Gamma$ such that for any subobject $U \subseteq \Gamma$

$$f^*U \leq I \qquad \Leftrightarrow \qquad U \leq \Pi_f I \ .$$

We then call $\Pi_f I$ the *dependent product* of I along f.

Remark 1.12. Once again, the dependent product is uniquely defined from its input data whenever it exists.

Definition 1.13. A Heyting category is a regular category with finite well-behaved subobject unions and dependent products.

Lemma 1.14. Let a pullback square $f' \circ g' = f \circ g$ in a Heyting category be given. Then we have

$$\Pi_{g'} g^*(X) = f'^* \Pi_f(X)$$

for any subobject X.

The Heyting categories are the domains for the models of our first order theories in these semantics. As Sets also is a Heyting category, this is a generalization of the notion of model as sets-with-structure.

Definition 1.15. A Σ-structure in a Heyting category $\mathcal{C}$ is an assignment of

- an object $[\![\sigma]\!] \in \mathcal{C}$ for each $\sigma \in \mathtt{sorts}\,\Sigma$
- a morphism $[\![f]\!] : [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!] \to [\![\sigma]\!]$ for each function symbol f of arity $(\vec{\sigma}, \sigma)$
- and a subobject $[\![\Xi]\!] \subseteq [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!]$ for each atomic proposition Ξ of arity $(\vec{\sigma})$.

Let us use the abbreviation $[\![x_1 : \sigma_1. \ldots . x_n : \sigma_n]\!] = [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!]$. Let us also denote the projection map from $[\![\Gamma. x : \sigma. \Delta]\!]$ to $[\![\sigma]\!]$ by $\pi_{[\![\sigma]\!]}$ Now we can interpret every judgement of the form

$$\Gamma \mid t : \sigma$$

as an arrow $[\![\Gamma]\!] \to [\![\sigma]\!]$ in the following way,

$$\begin{aligned} [\![\Gamma. x : \sigma. \Delta \mid x : \sigma]\!] &= \pi_{[\![\sigma]\!]} \\ [\![\Gamma \mid f(\vec{\tau}) : \sigma]\!] &= [\![f]\!] \circ [\![\Gamma \mid \vec{\tau} : \vec{\sigma}]\!] \end{aligned}$$

where $[\![\Gamma \mid \vec{\tau}.t : \vec{\sigma}.\sigma]\!]$ abbreviates

$$[[\![\Gamma \mid \vec{\tau} : \vec{\sigma}]\!], [\![\Gamma \mid t : \sigma]\!]] : [\![\Gamma]\!] \to [\![\vec{\sigma}]\!] \times [\![\sigma]\!].$$

Let us now denote the projection map from $[\![\Gamma.\, x : \sigma]\!]$ to $[\![\Gamma]\!]$ by π. We interpret the propositional judgements

$$\Gamma \mid \varphi \ \texttt{prop}$$

as a subobject of $[\![\Gamma]\!]$ in the following way,

$$\begin{aligned}
[\![\Gamma \mid \top \ \texttt{prop}]\!] &= [\![\Gamma]\!] \\
[\![\Gamma \mid \bot \ \texttt{prop}]\!] &= \bot_{[\![\Gamma]\!]} \\
[\![\Gamma \mid \varphi \wedge \psi \ \texttt{prop}]\!] &= [\![\Gamma \mid \varphi \ \texttt{prop}]\!] \cap [\![\Gamma \mid \psi \ \texttt{prop}]\!] \\
[\![\Gamma \mid \varphi \vee \psi \ \texttt{prop}]\!] &= [\![\Gamma \mid \varphi \ \texttt{prop}]\!] \cup [\![\Gamma \mid \psi \ \texttt{prop}]\!] \\
[\![\Gamma \mid \varphi \Rightarrow \psi \ \texttt{prop}]\!] &= \Pi_{[\![\Gamma \mid \varphi \ \texttt{prop}]\!]}([\![\Gamma \mid \psi \ \texttt{prop}]\!]) \\
[\![\Gamma \mid \exists x : \sigma \ \varphi \ \texttt{prop}]\!] &= \Sigma_\pi([\![\Gamma.\, x : \sigma \mid \varphi \ \texttt{prop}]\!]) \\
[\![\Gamma \mid \forall x : \sigma \ \varphi \ \texttt{prop}]\!] &= \Pi_\pi([\![\Gamma.\, x : \sigma \mid \varphi \ \texttt{prop}]\!]) \\
[\![\Gamma \mid t =_\sigma \tau \ \texttt{prop}]\!] &= \mathrm{Eq}([\![\Gamma \mid t : \sigma]\!], [\![\Gamma \mid \tau : \sigma]\!]) \\
[\![\Gamma \mid \Xi(\vec{\tau})\texttt{prop}]\!] &= [\![\Gamma \mid \vec{\tau} : \vec{\sigma}]\!]^*([\![\Xi]\!])
\end{aligned}$$

and we say that a *model* of $\Sigma, \mathcal{T}$ in a Heyting category $\mathcal{C}$ is a Σ-structure in $\mathcal{C}$ such that for every axiom

$$\mid \Theta \vdash \vartheta$$

in $\mathcal{T}$, we have that

$$\cap([\![\Theta]\!]) \leq [\![\vartheta]\!]$$

as subobjects of the terminal object. Here $\cap([\![\Theta]\!])$ denotes the subobject intersection of all $[\![\theta_i]\!]$.

The fact that $\mathbb{C}_{\Sigma,\mathcal{T}}$ is an initial object for these models can be stated in the following way.

Theorem 1.16. There is a bijection between the models $\mathcal{M}$ of Σ, $\mathcal{T}$ in $\mathcal{C}$ and the Heyting functors $F : \mathbb{C}_{\Sigma,\mathcal{T}} \to \mathcal{C}$.

Proof. See D1.4 of [4]. □

This can be understood as saying that $\mathbb{C}_{\Sigma,\mathcal{T}}$ is a standard interpretation of $\Sigma, \mathcal{T}$.

Remark 1.17. The syntactic category actually has an even stronger classifying property. One can define the notion of homomorphism between Σ-structures in a category $\mathcal{C}$. The homomorphisms between models then correspond to natural transformations between Heyting functors out of $\mathbb{C}_{\Sigma,\mathcal{T}}$. We will however not recover this desirable property on the type theoretic side.

1.2 Basics of type theory

Similar to first order logic, in type theory one has sequents of different kinds, where the left side contains variables together with information about what kind of values they take. They look like this.

$$x : A.\, y : B(x) \vdash C(x, y) \ \texttt{type}$$

Unlike first order logic one does not have a separation between concepts like sorts or propositions. Both these concepts are represented by 'types' which work uniformly. Note that one of the 'variable holders' of the context, $B(x)$, is dependent on x. This is an important feature of the type theories that we will investigate here: Their contexts can be very rich.

From the above sequent we can derive

$$\vdash x : A.\, y : B(x).\, z : C(x, y) \ \texttt{ctxt}$$

which indicates that the list $x : A.\, y : B(x).\, z : C(x, y)$ is a well-formed context. The idea is that all the things in the context that depend on a variable x are situated to the right of that variable, which allows one to formulate good rules for substitution. Suppose for instance that we have a term

$$x : A.\, y : B(x).\, z : C(x, y) \vdash t(x, y, z) : \alpha(x, y)$$

and that we have a term

$$x : A \vdash b(x) : B(x),$$

we can then substitute this term for y and get

$$x : A.\, z : C(x, b(x)) \vdash t(x, b(x), z) : \alpha(x, b(x)).$$

This sets type theories apart from first order logic. Another important difference is that we take a sort of equality judgement to be primitive to the type theory: We may have

$$x : A.\, z : C(x, b(x)) \vdash t(x, b(x), z) = s(x, z) : \alpha(x, b(x))$$

which essentially means that in this context, we may substitute the terms $t(x, b(x), z)$ and $s(x, z)$ for eachother. This is a bit different from the propositional equality for first order logic. The judgement $t(x, b(x), z) = s(x, z) : \alpha(x, b(x))$ is not represented by a type but is a primitive. Nothing stops you from adding a type for representing equality, though! This equality may have a behaviour that is wildly different from the judgemental equality present in the sequent above. This will however not be the case for the type theory studied here.

Similar to the first order logic is that we have a model of the type theory built from the syntax. The objects of this type theory are the contexts up to renaming of variables. This means that the contexts

$$\vdash x : A.\, z : C(x, b(x)) \ \texttt{ctxt} \qquad \vdash y : A.\, x : C(y, b(y)) \ \texttt{ctxt}$$

represent the same object. The morphisms are generated by the terms of the type theory, in the sense that we need to provide a term for each component of a context, but in a way that respects the way the context depends on itself. Let us for instance say that we have

$$\vdash x : A.\, z : C(x, b(x)) \ \texttt{ctxt} \qquad \vdash y : D.\, w : E(y) \ \texttt{ctxt},$$

let us call them Γ' and Γ for brevity. To provide a context morphism $\Gamma' \to \Gamma$ we need to provide first a term

$$x : A.\, z : C(x, b(x)) \vdash r(x, z) : D.$$

Then we also need to provide a term of E, but in order to respect the way the context depends on itself, we need it to be $E(r(x, z))$,

$$x : A.\, z : C(x, b(x)) \vdash h(x, z) : E(r(x, z)).$$

This generates a context morphism. They are identified up to judgemental equality, which for another context morphism $r'(x, z).h'(x, z)$ would mean that we have the judgements

$$x : A.\, z : C(x, b(x)) \vdash r(x, z) = r'(x, z) : D$$

$$x : A.\, z : C(x, b(x)) \vdash h(x, z) = h'(x, z) : E(r(x, z)).$$

Note that a priori we would not even expect the judgement $h'(x, z) : E(r(x, z))$ to typecheck (only $h' : E(r'(x, z))$) but in the presence of the judgement $r(x, z) = r'(x, z) : D$ it works out.

2 Syntax

This section we will introduce and investigate the syntax of a type theory $\mathscr{FOT}(\Sigma, \mathcal{T})$ for a first order theory $\mathcal{T}$ over a signature Σ. The main results that we wish to prove are the following.

- The term model $\mathcal{I}_{\Sigma,\mathcal{T}}$ is equivalent to the syntactic category $\mathbb{C}_{\Sigma,\mathcal{T}}$.
- This equivalence is a Heyting equivalence.

This way we establish that Heyting functors out of $\mathcal{I}_{\Sigma,\mathcal{T}}$ are the same thing as Heyting functors out of $\mathbb{C}_{\Sigma,\mathcal{T}}$. We will not be explicit about ensuring that the equivalence that we construct is a Heyting equivalence, but rather only provide the Heyting structure on $\mathcal{I}_{\Sigma,\mathcal{T}}$.

The way that we go about constructing this equivalence is that we find translations back and forth between $\mathscr{FOT}(\Sigma, \mathcal{T})$ and the first order theory $\mathcal{T}$ over Σ. The main obstacle is the appearance of *definite descriptions* in the propositions of $\mathscr{FOT}(\Sigma, \mathcal{T})$, in trying to translate into first order logic. The goal of Section 2.3 is to show how these definite descriptions may be eliminated from the syntax for the propositional judgements in definite description free contexts.

We subsequently factor the translation into some intermediate stages, each of which straightforwardly yields an equivalence of categories.

2.1 First order logic as a type theory

First order logic has sorts, which describe how the *terms* of the language fit together. It also has *propositions* or *predicates* which are used to express facts about its terms. In type theory one does not make such a distinction between the sorts and the propositions. Rather, one represents the truth of a proposition by a proof-term, where there are potentially distinct proof-terms of the same proposition. These proof-terms carry information about how the truth of the proposition was derived. No such information is retained in first order logic.

The definition we give here is within framework provided by [7], where a general method for constructing a type theory is given. A common feature of these type theories is that they have the *weakening* and *substitution* rules.

The subsection that follows is essentially one long definition where all the rules of the type theory are presented. First, let us look at our judgement forms,

$$\Gamma\ \mathtt{ctxt} \qquad \varphi\ \mathtt{prop} \qquad \sigma\ \mathtt{sort} \qquad \rho : \varphi \qquad t : \sigma$$

which are the ones that we will be devoting the most attention to, but we also have judgement forms for equalities.

$$\Gamma = \Delta\ \mathtt{ctxt} \qquad \varphi = \psi\ \mathtt{prop} \qquad \sigma = \sigma'\ \mathtt{sort} \qquad \rho = \delta : \varphi \qquad t = \tau : \sigma$$

The judgement forms for contexts will only take an empty context as a primitive, although we will derive how to consider contexts over other contexts. The equality judgement form for sorts is an artefact from the framework we are using (two sorts will only be judged equal in some context if the are syntactically identical to begin with). We will also use the abbreviation α `type` to signify that the same rule applies regardless of whether α is judged to be a proposition or a sort.

Given a first-order signature Σ and theory $\mathcal{T}$ over Σ, let us begin defining the type theory $\mathscr{FOT}(\Sigma, \mathcal{T})$. First we define what the sequents of the type theory are (via the raw syntax) and then we introduce the rules, by which we single out the derivable sequents.

2.1.1 Raw syntax

A first-order signature consists of a set of sort symbols, $\mathtt{sorts}\,\Sigma$, a set of function symbols $\mathtt{functions}_n\Sigma$ and a set of atomic propositional symbols $\mathtt{atoms}_n\Sigma$, the latter two of which can be graded by the number of arguments they take.

Definition 2.1. The following clauses define the raw syntax of $\mathscr{FOT}(\Sigma, \mathcal{T})$. Quantifiers like $\forall\, x : \sigma\ \varphi$ and variable bindings like $x.\tau$ indicate that x is now a bound variable of the entire expression and the clause only applies if x is not bound in φ or τ respectively. We will also not distinguish between renaming of bound variables. Also, every axiom of the theory

$$\mid \Theta \vdash \vartheta$$

can be graded by the number of formulas in Θ.
Clauses for sorts

$$\sigma \equiv \quad A \quad (A \in \mathtt{sorts}\,\Sigma)$$

Clauses for raw sort-terms

$$\begin{aligned} t, \tau \equiv \quad & x, y \text{ (sort-variables)} \mid \imath\, x : \sigma\, \varphi\, (\rho, \delta) \\ & \mid f(\vec{\tau}) \quad (f \in \mathtt{functions}_n\Sigma,\ \vec{\tau}\ n\text{-tuple of raw sort-terms}) \end{aligned}$$

Clauses for raw proof-terms

$$\begin{aligned} \rho, \delta, \pi \quad \equiv \quad & p, q \text{ (proof-variables)} \mid {=}\mathtt{I}(\sigma, t) \\ & \mid \vee\mathtt{I}_L(\varphi, \psi, \rho) \mid \vee\mathtt{I}_R(\varphi, \psi, \rho) \mid \vee\mathtt{E}(\varphi, \psi, \vartheta, p.\rho, q.\delta, \pi) \\ & \mid \exists\,\mathtt{I}(\sigma, x.\psi, t, \rho) \mid \exists\,\mathtt{E}(\sigma, x.\psi, \vartheta, \rho, x.p.\delta) \\ & \mid \forall\mathtt{I}(\sigma, x.\psi, x.\rho) \mid \forall\mathtt{E}(\sigma, x.\psi, \rho, t) \\ & \mid \exists\,\mathtt{I}(\varphi, x.\varphi, t, \rho) \mid \exists\,\mathtt{E}(\varphi, p.\psi, \vartheta, \rho, p.q.\delta) \\ & \mid \forall\mathtt{I}(\varphi, p.\psi, p.\rho) \mid \forall\mathtt{E}(\varphi, p.\psi, \rho, \delta) \\ & \mid \partial\, x : \sigma\, \varphi\, (\rho, \delta) \\ & \mid \mathcal{A}(\vec{\rho})\ (\mathcal{A} \in \mathtt{axioms}\, \mathcal{T}_n) \end{aligned}$$

Clauses for raw formulas

$$\begin{aligned}\varphi, \psi, \vartheta \equiv \quad & \top \mid \bot \mid \varphi \vee \psi \\ & \mid \exists x : \sigma \; \psi \mid \forall x : \sigma \; \psi \\ & \mid \exists p : \varphi \; \psi \mid \forall p : \varphi \; \psi \\ & \mid t =_\sigma \tau \\ & \mid \Xi(\vec{\tau}) \;\; (\Xi \in \mathtt{atoms}_n \Sigma,\; \vec{\tau} \text{ } n\text{-tuple of raw sort-terms})\end{aligned}$$

Expressions of the form

$$\imath\, x : \sigma\, \varphi\, (\rho, \delta) \quad \text{and} \quad d\, x : \sigma\, \varphi\, (\rho, \delta)$$

are the terms for definite descriptions, representing the thing described by φ and the canonical proof that it satisfies φ respectively, and are said to be $\imath$-expressions. Expressions which contain no $\imath$-term as a subexpression are said to be $\imath$-free.

Given that we now have all the raw symbols of the type theory, we can define the sequents as being expressions of the form

$$\Gamma \vdash \mathcal{J}$$

where Γ is an expression like $x : \alpha. \ldots. w : \varphi$ where each entry is a raw formula or sort α with a variable x of the appropriate kind such that all the variables are distinct, and $\mathcal{J}$ is one of the judgement forms

$$\Gamma\ \mathtt{ctxt} \qquad \varphi\ \mathtt{prop} \qquad \sigma\ \mathtt{sort} \qquad \rho : \varphi \qquad t : \sigma$$

or

$$\Gamma = \Delta\ \mathtt{ctxt} \qquad \varphi = \psi\ \mathtt{prop} \qquad \sigma = \sigma'\ \mathtt{sort} \qquad \rho = \delta : \varphi \qquad t = \tau : \sigma.$$

2.1.2 Rules for symbols from the signature

These are the rules for the function symbols and atomic formulas of a signature Σ of first order logic. For each sort σ of Σ we introduce a judgement

$$\frac{\Gamma\ \mathtt{ctxt}}{\Gamma \vdash \sigma\ \mathtt{sort}}$$

and for each function symbol $f : \vec{\sigma} \to \sigma$ we introduce a term of arity $\vec{\sigma}$ with the introduction rule

$$\frac{\Gamma \vdash \tau_1 : \sigma_1 \qquad \ldots \qquad \Gamma \vdash \tau_n : \sigma_n}{\Gamma \vdash f(\vec{\tau}) : \sigma}$$

For each atomic formula Ξ with formula arity $\vec{\sigma}$ we give the introduction rule

$$\frac{\Gamma \vdash \tau_1 : \sigma_1 \qquad \ldots \qquad \Gamma \vdash \tau_n : \sigma_n}{\Gamma \vdash \Xi(\vec{\tau})\ \mathtt{prop}}$$

These are the rules that give us access to the symbols of the signature.

2.1.3 Logical and structural rules

Now for the lion part of the rules. Note that the diamond shape $\Diamond$ is a placeholder for the length 0 context as defined in the raw syntax.

Context rules:

$$\frac{}{\vdash \Diamond\ \texttt{ctxt}} \qquad \frac{\vdash \Gamma\ \texttt{ctxt} \qquad \Gamma \vdash \alpha\ \texttt{type}}{\vdash \Gamma.\, x : \alpha\ \texttt{ctxt}} \qquad \frac{\vdash \Gamma.\, x : \alpha.\, \Delta\ \texttt{ctxt}}{\Gamma.\, x : \alpha.\, \Delta \vdash x : \alpha}$$

Structural rules:

$$\textsc{Substitution}\ \frac{\Gamma \vdash \alpha\ \texttt{type} \qquad \Gamma.\, \Delta \vdash \mathcal{J}}{\Gamma.\, x : \alpha.\, \Delta \vdash \mathcal{J}} \qquad \textsc{Weakening}\ \frac{\Gamma \vdash t : \alpha \qquad \Gamma.\, x : \alpha.\, \Delta \vdash \mathcal{J}}{\Gamma.\, \Delta\,[x \backslash t] \vdash \mathcal{J}\,[x \backslash t]}$$

The terms associated to the type given by a first order formula φ are to be understood as proofs of φ and for now we don't distinguish between different proofs but take them to be the same, as stated in the following equality rule.

$$\textsc{Proof irrelevance}\ \frac{\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \rho : \varphi \qquad \Gamma \vdash \delta : \varphi}{\Gamma \vdash \rho = \delta : \varphi}$$

We describe the type theoretic judgements corresponding to the different logical rules. First the truth and false:

$$\frac{\vdash \Gamma\ \texttt{ctxt}}{\Gamma \vdash \top\ \texttt{prop}} \qquad \frac{\vdash \Gamma\ \texttt{ctxt}}{\Gamma \vdash \top\texttt{I} : \top} \qquad \frac{\vdash \Gamma\ \texttt{ctxt}}{\Gamma \vdash \bot\ \texttt{prop}} \qquad \frac{\Gamma \vdash \rho : \bot \qquad \Gamma \vdash \varphi\ \texttt{prop}}{\Gamma \vdash \bot\texttt{E}(\varphi, \rho) : \varphi}$$

Equality:

$$\frac{\Gamma \vdash \sigma\ \texttt{sort} \qquad \Gamma \vdash t : \sigma \qquad \Gamma \vdash \tau : \sigma}{\Gamma \vdash t =_\sigma \tau\ \texttt{prop}} \qquad \frac{\Gamma \vdash \sigma\ \texttt{sort} \qquad \Gamma \vdash t : \sigma}{\Gamma \vdash {=}\texttt{I}(\sigma, \tau) : t =_\sigma t}$$

$$\frac{\Gamma \vdash \sigma\ \texttt{sort} \qquad \Gamma \vdash t : \sigma \qquad \Gamma \vdash \tau : \sigma \qquad \Gamma \vdash \rho : t =_\sigma \tau}{\Gamma \vdash t = \tau : \sigma}$$

Disjunction:

$$\frac{\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \psi\ \texttt{prop}}{\Gamma \vdash \varphi \vee \psi\ \texttt{prop}}$$

$$\frac{\begin{array}{c}\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \psi\ \texttt{prop} \\ \Gamma \vdash \rho : \varphi\end{array}}{\Gamma \vdash \vee\texttt{I}_L(\varphi, \psi, \rho) : \varphi \vee \psi} \qquad \frac{\begin{array}{c}\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \psi\ \texttt{prop} \\ \Gamma \vdash \rho : \psi\end{array}}{\Gamma \vdash \vee\texttt{I}_R(\varphi, \psi, \rho) : \varphi \vee \psi}$$

$$\frac{\begin{array}{c}\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \psi\ \texttt{prop} \qquad \Gamma \vdash \vartheta\ \texttt{prop} \\ \Gamma \vdash \pi : \varphi \vee \psi \qquad \Gamma.\, p : \varphi \vdash \rho : \vartheta \qquad \Gamma.\, q : \psi \vdash \delta : \vartheta\end{array}}{\Gamma \vdash \vee\texttt{E}(\varphi, \psi, \vartheta, \pi, p.\rho, q.\delta) : \vartheta}$$

Existential quantification for sorts:

$$\frac{\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop}}{\Gamma \vdash \exists\, x : \sigma\ \psi\ \mathtt{prop}}$$

$$\frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \\ \Gamma \vdash t : \sigma \qquad \Gamma \vdash \delta : \psi\,[\,x \backslash t\,]\end{array}}{\Gamma \vdash \exists\,\mathtt{I}(\sigma, x.\psi, t, \delta) : \exists\, x : \sigma\ \psi}$$

$$\frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \qquad \Gamma \vdash \vartheta\ \mathtt{prop} \\ \Gamma \vdash \rho : \exists\, x : \sigma\ \psi \qquad \Gamma.\, x : \sigma.\, p : \psi \vdash \delta : \vartheta\end{array}}{\Gamma \vdash \exists\,\mathtt{E}(\sigma, x.\psi, \vartheta, \rho, x.p.\delta) : \vartheta}$$

Universal quantification for sorts:

$$\frac{\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop}}{\Gamma \vdash \forall\, x : \sigma\ \psi\ \mathtt{prop}} \qquad \frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \\ \Gamma.\, x : \sigma \vdash \rho : \psi\end{array}}{\Gamma \vdash \forall\mathtt{I}(\sigma, x.\psi, x.\rho) : \forall\, x : \sigma\ \psi}$$

$$\frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \\ \Gamma \vdash \rho : \forall\, x : \sigma\ \psi \qquad \Gamma \vdash t : \sigma\end{array}}{\Gamma \vdash \forall\mathtt{E}(\sigma, x.\psi, \rho, t) : \psi\,[\,x \backslash t\,]}$$

The most drastic departure from the standard components of first order logic is a term which can be introduced by supplying a proof of existence and uniqueness of a variable satisfying some predicate, i.e., a definite description operator.

$$\frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \\ \Gamma \vdash \varepsilon : \exists\, x : \sigma\ \psi \qquad \Gamma.\, x : \sigma.\, p : \psi.\, y : \sigma.\, q : \psi\,[\,x \backslash y\,] \vdash \upsilon : x =_\sigma y\end{array}}{\Gamma \vdash \imath\ x : \sigma\ \psi(\varepsilon, x.p.y.q.\upsilon) : \sigma}$$

$$\frac{\begin{array}{c}\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \psi\ \mathtt{prop} \\ \Gamma \vdash \varepsilon : \exists\, x : \sigma\ \psi \qquad \Gamma.\, x : \sigma.\, p : \psi.\, y : \sigma.\, q : \psi\,[\,x \backslash y\,] \vdash \upsilon : x =_\sigma y\end{array}}{\Gamma \vdash \rotatebox[origin=c]{180}{$\imath$}\ x : \sigma\ \psi(\varepsilon, x.p.y.q.\upsilon) : \psi\,[\,x \backslash \imath\ x : \sigma\ \psi(\varepsilon, x.p.y.q.\upsilon)\,]}$$

Because we get sort-terms which depend on proof-terms in this type theory and because propositions in first order logic are formed from sort-terms, the formulas corresponding to a conjunction or implication will have the second argument depend on the first one. This is not needed in standard first order logic as there sort-terms cannot be formed from proof-terms. Instead of conjunction and implication we will call them existential quantification and universal quantification but for propositions. Rules for existential quantification for propositions:

$$\frac{\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop}}{\Gamma \vdash \exists\, p : \varphi\ \psi\ \texttt{prop}}$$

$$\frac{\begin{array}{cc}\Gamma \vdash \varphi\ \texttt{prop} & \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop} \\ \Gamma \vdash \rho : \varphi & \Gamma \vdash \delta : \psi\,[\,p \backslash \rho\,]\end{array}}{\Gamma \vdash \exists\,\texttt{I}(\varphi, p.\psi, \rho, \delta) : \exists\, p : \varphi\ \psi}$$

$$\frac{\begin{array}{c}\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop} \qquad \Gamma \vdash \vartheta\ \texttt{prop} \\ \Gamma \vdash \rho : \exists p : \varphi\ \psi \qquad \Gamma.\, p : \varphi.\, q : \psi \vdash \delta : \vartheta\end{array}}{\Gamma \vdash \exists\,\texttt{E}(\varphi, p.\psi, \vartheta, \rho, p.q.\delta) : \vartheta}$$

Universal quantification for propositions:

$$\frac{\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop}}{\Gamma \vdash \forall\, p : \varphi\ \psi\ \texttt{prop}} \qquad \frac{\begin{array}{c}\Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop} \\ \Gamma.\, p : \varphi \vdash \rho : \psi\end{array}}{\Gamma \vdash \forall\texttt{I}(\varphi, p.\psi, p.\rho) : \forall\, p : \varphi\ \psi}$$

$$\frac{\begin{array}{cc}\Gamma \vdash \varphi\ \texttt{prop} & \Gamma.\, p : \varphi \vdash \psi\ \texttt{prop} \\ \Gamma \vdash \rho : \forall\, p : \varphi\ \psi & \Gamma \vdash \delta : \varphi\end{array}}{\Gamma \vdash \forall\texttt{E}(\varphi, p.\psi, \rho, \delta) : \psi\,[\,p \backslash \delta\,]}$$

Note that we have not shown the equality rules here. There is an equality rule for each argument of a symbol, making sure that if we provide two arguments that are judged equal by the type theory then the result is judged equal as well. There are also rules for making sure that judgemental equality is an equivalence relation, and that anytime two expressions are judged equal, one may replace one by the other in sequents. See [7] for the full definition.

2.1.4 Rules for axiom terms

Now that we have all the rules for deriving propositions, we will add the symbol rules for the proof-terms witnessing the axioms.

$$\frac{\Gamma \vdash \vec{\tau} : \vec{\sigma} \qquad \Gamma \vdash \rho_1 : \theta_1(\vec{\tau}) \qquad \ldots \qquad \Gamma \vdash \rho_n : \theta_n(\vec{\tau})}{\Gamma \vdash \mathcal{A}(\vec{\tau}, \vec{\rho}) : \varphi(\vec{\tau})}$$

We note here that the formulas $\theta_i(\vec{\tau})$ and $\varphi(\vec{\tau})$ are not precisely the same as the assumptions and conclusion of an axiom of $\mathcal{T}$

$$\Delta \mid \Theta \vdash \varphi$$

but we have replaced conjunction and implication by prop-existential quantification and implication by prop-universal quantification.

2.2 Some syntactic conveniences

Here we will provide some syntactic sugar and prove that $\mathcal{I}_{\Sigma,\mathcal{T}}$ is a Heyting category.

In practice we will be a little more verbose when specifying the variable bindings of terms than needed. For instance, instead of writing

$$\exists\,\mathtt{E}(\varphi, p.\psi, \vartheta, \rho, p.q.\delta)$$

we would rather write

$$\begin{aligned}&\exists\,\mathtt{E}\{\\ &\qquad p : \varphi \vdash \psi,\\ &\qquad \vartheta,\\ &\qquad \rho,\\ &\qquad p : \varphi.\, q : \psi \vdash \delta\\ &\}\end{aligned}$$

or something of the sort. We will also seldom work directly with quantification over a sort or proposition, rather working with quantification over a context. First we introduce some notation for basic judgements with contexts. If $\Gamma.\,\Delta$ is a derivable context we will sometimes write

$$\Gamma \vdash \Delta\ \mathtt{ctxt}$$

and we will define context morphisms between contexts over Γ by induction on the context lengths

$$\frac{\Gamma \vdash \Delta_1\ \mathtt{ctxt} \qquad \Gamma \vdash \Delta_2.\, y : \varphi\ \mathtt{ctxt} \qquad \Gamma \vdash f : \Delta_1 \to \Delta_2 \qquad \Gamma.\,\Delta_1 \vdash \tau : \varphi\,[f]}{\Gamma \vdash f.\tau : \Delta_1 \to \Delta_2.\, y : \varphi}$$

where the substitution of a judgement $\mathcal{J}$ along a context morphism is given by

$$\frac{\Gamma \vdash \Delta_1, \Delta_2.\, y : \varphi\ \mathtt{ctxt} \qquad \Gamma.\,\Delta_2.\, y : \varphi \vdash \mathcal{J} \qquad \Gamma \vdash f.\tau : \Delta_1 \to \Delta_2.\, y : \varphi}{\Gamma.\,\Delta_1 \vdash \mathcal{J}\,[f]\,[\,y \backslash \tau\,]}$$

and we also denote $\Gamma \vdash f : \Diamond \to \Delta$ by

$$\Gamma \vdash f : \Delta.$$

Let us now pack together the fact that if two compound formulas of the same connective have equivalent subformulas, then they themselves are equivalent.

Proposition 2.2. For each logical connective $\bigcirc$ (including quantifiers) and pairs of formulas φ_1, φ_2 and ψ_1, ψ_2 approperiate for the connective such that

$$\Gamma \vdash \varphi_1\ \mathtt{prop} \qquad\qquad \Gamma \vdash \psi_1\ \mathtt{prop}$$

are derivable and equivalent in the sense that we have $\mathtt{i}_1, \mathtt{j}_1$ such that

$$\Gamma \vdash \mathtt{i}_1 : \varphi_1 \to \psi_1 \qquad\qquad \Gamma \vdash \mathtt{j}_1 : \psi_1 \to \varphi_1$$

are derivable, and the corresponding thing holds for φ_2, ψ_2 with $\mathtt{i}_2, \mathtt{j}_2$ (the precise formulation depends on the connective), we can form terms

$$\mathtt{i}\langle\bigcirc\rangle(\mathtt{i}_1, \mathtt{j}_1, \mathtt{i}_2, \mathtt{j}_2) \quad \text{and} \quad \mathtt{j}\langle\bigcirc\rangle(\mathtt{i}_1, \mathtt{j}_1, \mathtt{i}_2, \mathtt{j}_2)$$

such that the following are derivable

$$\Gamma \vdash \mathtt{i}\langle\bigcirc\rangle(\mathtt{i}_1, \mathtt{j}_1, \mathtt{i}_2, \mathtt{j}_2) : \varphi_1 \bigcirc \varphi_2 \to \psi_1 \bigcirc \psi_2,$$

$$\Gamma \vdash \mathtt{j}\langle\bigcirc\rangle(\mathtt{i}_1, \mathtt{j}_1, \mathtt{i}_2, \mathtt{j}_2) : \psi_1 \bigcirc \psi_2 \to \varphi_1 \bigcirc \varphi_2.$$

Similarly for sort-quantification we can construct equivalences for $Q\ x : \sigma\ \varphi$ and $Q\ x : \sigma\ \psi$ from an equivalence for φ, ψ.

Proof. A matter of using the introduction and elimination rules for each connective $\bigcirc$. $\square$

We also have type inference. We will state this the following way, and only need it for proof-terms.

Proposition 2.3. We can define the formula $\mathtt{p}(\rho)$ of a proof-term ρ over Γ called the *inferred type of* ρ such that if

$$\Gamma \vdash \rho : \vartheta$$

is derivable, then so is

$$\Gamma \vdash \rho : \mathtt{p}(\rho).$$

Proof. Both the definition and proof proceed by case analysis on ρ. If ρ is a variable, just take $\mathtt{p}(\rho)$ to be the formula in the context that it came from.

If ρ is a logical symbol, the outermost part of ρ contains all he information needed to infer the type, and admissability guarantees that we can derive $\Gamma \vdash \rho : \mathtt{p}(\rho)$. For instance, if ρ is existential introduction, we have

$$\exists\mathtt{I}(\ y : \sigma \vdash \varphi,\ t,\ \delta\ )$$

from which we take $\mathtt{p}(\rho) = \exists\, y : \sigma\ \varphi$ and if we have

$$\Gamma \vdash \exists\mathtt{I}(\ y : \sigma \vdash \varphi,\ t,\ \delta\ ) : \vartheta$$

then by admissibility we also have

$$\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, y : \sigma \vdash \varphi\ \mathtt{prop} \qquad \Gamma \vdash t : \sigma \qquad \Gamma \vdash \delta : \varphi\,[\,x \backslash t\,]$$

which allows us to apply existential introduction and get

$$\Gamma \vdash \exists\mathtt{I}(\ y : \sigma \vdash \varphi,\ t,\ \delta\ ) : \exists\, y : \sigma\ \varphi.$$

If ρ is an axiom term $\mathcal{A}(\vec{\rho})$, we take $\mathtt{p}(\rho)$ to be the conclusion of the axiom. By admissibility we get

$$\Gamma \vdash \rho_i : \theta_i$$

for each i if $\Gamma \vdash \mathcal{A}(\vec{\rho}) : \vartheta$, then we apply the symbol rule for the axiom term and get the desired conclusion. $\square$

Let us state some propositions which we will not properly prove but subsequently use.

Proposition 2.4. If $\Gamma \vdash \varphi = \psi$ $\mathtt{prop}$ then φ and ψ come from the same term constuctor, and their arguments are judged equal too.

Proof. Essentially, the only equality rules for propositions are those from the symbol rules, where the propositions are judged equal if one can "substitute" in equal subterms (not necessarily substitute in the sense of applying substitution rule in the type theory, but in the sense that φ and ψ have been substituted into $\varphi \vee \vartheta$ and $\psi \vee \vartheta$). $\square$

Proposition 2.5. If $\Gamma \vdash \rho : \varphi$ and $\Gamma \vdash \rho : \psi$ then $\Gamma \vdash \varphi = \psi$ $\mathtt{prop}$.

For the quantifications over contexts, we have the following

$$\frac{\Gamma \vdash \Delta \ \mathtt{ctxt} \qquad \Gamma.\,\Delta \vdash \psi \ \mathtt{prop}}{\Gamma \vdash \exists\,\Delta\ \psi \ \mathtt{prop}}$$

where $\exists\,\Delta\ \psi$ is defined by generating cases

$$\exists\, x : \sigma.\,\Delta\ \psi = \exists\, x : \sigma\,\exists\,\Delta\ \psi \quad \text{and} \quad \exists\, p : \varphi.\,\Delta\ \psi = \exists\, p : \varphi\,\exists\,\Delta\ \psi$$

and similar for universal quantification. We can straightforwardly define proof-terms using the generating cases satisfying

$$\frac{\Gamma \vdash \Delta \ \mathtt{ctxt} \qquad \Gamma.\,\Delta \vdash \psi \ \mathtt{prop} \qquad \Gamma \vdash f : \Delta \qquad \Gamma \vdash \rho : \psi\,[f]}{\Gamma \vdash \exists\mathtt{I}(\,\Delta \vdash \psi,\, f,\, \rho\,) : \exists\,\Delta\ \psi}$$

$$\frac{\begin{array}{c}\Gamma \vdash \Delta \ \mathtt{ctxt} \qquad \Gamma.\,\Delta \vdash \psi \ \mathtt{prop} \qquad \Gamma \vdash \vartheta \ \mathtt{prop} \\ \Gamma \vdash \rho : \exists\,\Delta\ \psi \qquad \Gamma.\,\Delta.\,p : \psi \vdash \delta : \vartheta\end{array}}{\Gamma \vdash \exists\mathtt{E}(\,\Delta \vdash \psi,\, \vartheta,\, \rho,\, \Delta.\,p : \psi \vdash \delta\,)}$$

and similarly for universal quantification we can define proof-terms by replacing the sort or proposition being quantified over by a context.

Also, we can extend propositional equality to contexts by

$$(f.\rho =_{\Gamma.\,p:\varphi} g.\delta) = (f =_\Gamma g) \qquad (f.t =_{\Gamma.\,x:\sigma} g.\tau) = (\exists\, p : t =_\sigma \tau)\ \ f =_\Gamma g$$

with base case $f =_\diamond g = \top$. It has

$$\frac{\Gamma \vdash \Delta \ \mathtt{ctxt} \qquad \Gamma \vdash f : \Delta \qquad \Gamma \vdash g : \Delta}{\Gamma \vdash f =_\Delta g \ \mathtt{prop}} \qquad \frac{\Gamma \vdash \Delta \ \mathtt{ctxt} \qquad \Gamma \vdash f : \Delta}{\Gamma \vdash {=}\mathtt{I}(\Delta, f) : f =_\Delta f}$$

and also satisfies

$$\frac{\Gamma \vdash \Delta \ \texttt{ctxt} \qquad \Gamma \vdash f, g : \Delta \qquad \Gamma \vdash \rho : f =_\Delta g}{\Gamma \vdash f = g}$$

which allows us to extend the definite descriptions by the rules

$$\frac{\begin{array}{c}\Gamma \vdash \Delta \ \texttt{ctxt} \qquad \Gamma.\,\Delta \vdash \varphi \ \texttt{prop} \qquad \Gamma \vdash \varepsilon : \exists\, \Delta \top \\ \Gamma.\, \mathsf{x} : \Delta.\, p : \varphi\,[\mathsf{x}].\, \mathsf{y} : \Delta.\, \varphi\,[\mathsf{y}] \vdash \upsilon : \mathsf{x} =_\Delta \mathsf{y}\end{array}}{\Gamma \vdash \imath\ \Delta\ \varphi(\varepsilon, \upsilon) : \Delta}$$

$$\frac{\begin{array}{c}\Gamma \vdash \Delta \ \texttt{ctxt} \qquad \Gamma.\,\Delta \vdash \varphi \ \texttt{prop} \qquad \Gamma \vdash \varepsilon : \exists\, \Delta \top \\ \Gamma.\, \mathsf{x} : \Delta.\, p : \varphi\,[\mathsf{x}].\, \mathsf{y} : \Delta.\, \varphi\,[\mathsf{y}] \vdash \upsilon : \mathsf{x} =_\Delta \mathsf{y}\end{array}}{\Gamma \vdash \eth\ \Delta\ \varphi(\varepsilon, \upsilon) : \varphi\,[\imath\ \Delta\ \varphi(\varepsilon, \upsilon)]}$$

2.2.1 Heyting structure on term model

We now show that $\mathcal{I}_{\Sigma,\mathcal{T}}$ has the structure of a Heyting category. First note that we get the factorization into regular epi followed by mono by factorizing $f : \Gamma \to \Delta$ as

$$f.\rho : \Gamma \to \Delta.\, p : \exists\,\Gamma\ f =_\Delta \mathsf{y} \qquad\qquad \mathsf{y} : \Delta.\, p : \exists\,\Gamma\ f =_\Delta \mathsf{y} \to \Gamma$$

where y are the variables in Δ. The term ρ is given by existential introduction on the equality intro on $(f =_\Delta \mathsf{y}\,[f]) = f =_\Delta f$.

Proposition 2.6. The above factorization is a regular epi followed by a mono.

Proof. The latter arrow is a monomorphism by proof irrelevance. To show that the former is a regular epi, take some other factorization

$$f = g \circ h : \Gamma \to \Delta' \to \Delta$$

and construct the arrow $\Delta.\, p : \exists\,\Gamma\ f =_\Delta \mathsf{y} \to \Delta'$ by the following observation:

If $g : \Delta' \to \Delta$ is mono then there is a proof-term δ such that

$$\Delta.\, \mathsf{v} : \Delta'.\, \mathsf{w} : \Delta'.\, p : g\,[\mathsf{v}] =_\Delta g\,[\mathsf{w}] \vdash \delta : \mathsf{v} =_{\Delta'} \mathsf{w}$$

is derivable. This means that there is a proof-term υ such that

$$\Delta.\, \mathsf{v} : \Delta'.\, p : g\,[\mathsf{v}] =_\Delta \mathsf{y}.\, \mathsf{w} : \Delta'.\, q : g\,[\mathsf{w}] =_\Delta \mathsf{y} \vdash \upsilon : \mathsf{v} = \mathsf{w}$$

and we can from $p : \exists\,\Gamma\ f =_\Delta \mathsf{y}$ get $\varepsilon : \exists\,\Delta'\ g =_\Delta \mathsf{y}$ by doing an existential elimination on p to get some $\mathsf{x} : \Gamma$ such that $f(\mathsf{x}) =_\Delta \mathsf{y}$. The factorization $f = g \circ h$ means that we can do existential introduction on $h(\mathsf{x})$ together with the given proof that $g \circ h(\mathsf{x}) =_\Delta \mathsf{y}$. This makes the following diagram commute with $t = \imath\ \Delta'\ (g =_\Delta \mathsf{y})(\varepsilon, \upsilon)$,

$$\begin{tikzcd}
 & \Delta' \arrow[dr, tail, "g"] & \\
\Gamma \arrow[ur, "h"] \arrow[r, "f.\rho"'] & \Delta.\, p : \exists\,\Gamma\ f =_\Delta \mathsf{y} \arrow[u, "t"] \arrow[r, tail, "\mathsf{y}"'] & \Delta
\end{tikzcd}.$$

$\square$

Remark 2.7. This does not only give us a regular factorization on $\mathcal{I}_{\Sigma,\mathcal{T}}$, it also gives us that any monomorphism can be represented by a proposition, hence any subobject can be represented by a proposition. This representation can then be exploited to construct the subobject intersections, unions, dependent sums and dependent products using our logical connectives, see Johnstone [4, D1.4]. We will list them below.

Let $f : \Gamma' \to \Gamma$ be an arrow of the term model, where Γ' has variables x and Γ has variables y.

- The terminal object is given by $\vdash \lozenge$ `ctxt`.
- The objects $\vdash \Gamma$ `ctxt` and $\vdash \Gamma'$ `ctxt` have the product given by $\vdash \Gamma. \Gamma'$ `ctxt`.
- Two arrows $f, g : \Gamma' \to \Gamma$ has equalizer given by $\Gamma' \vdash f(\mathsf{x}) =_\Gamma g(\mathsf{x})$ `prop`.
- The initial subobjects are given by $\Gamma \vdash \bot$ `prop`.
- The subobject union of $\Gamma \vdash \varphi$ `prop` and $\Gamma \vdash \psi$ `prop` is given by $\Gamma \vdash \varphi \vee \psi$ `prop`.
- The dependent sum of a subobject $\Gamma' \vdash \varphi$ `prop` along f is given by $\Gamma \vdash \exists\, \Gamma'\ f(\mathsf{x}) = \mathsf{y}$.
- The dependent product of a subobject $\Gamma' \vdash \varphi$ `prop` along f is given by $\Gamma \vdash \forall\, \Gamma'\ f(\mathsf{x}) = \mathsf{y}$.

2.3 Translating judgements into definite description free fragment

The conservativity of definite descriptions over first order logic has already been established, see for example Fourman [1] for a topos theoretic perspective. The goal of this section is to prove a "type theoretic" conservativity of the definite descriptions in order to get an equivalence of categories between the term model $\mathcal{I}_{\Sigma,\mathcal{T}}$ and the syntactic category $\mathbb{C}_{\Sigma,\mathcal{T}}$.

To translate into a definite description free fragment of the type theory, we will begin by translating judgements in $\imath$-free contexts. This definition will be made by induction on the structure of the judgement, i.e., the translation will built by first translating the subexpressions and then putting them together in an appropriate way.

The main goal of the translation is that we translate a sequent like

$$\Gamma \vdash \varphi \ \texttt{prop}$$

into one like

$$\Gamma \vdash \texttt{t}\,(\varphi) \ \texttt{prop}$$

where $\texttt{t}\,(\varphi)$ is $\imath$-free, and the second sequent is derivable whenever the first one is. To prove such a thing one usually has to do an induction on trees of derivation, i.e., go through all the rules of the type theory, assume that it works for the premises of the rule and use that to prove that it works for the conclusion as well.

To do get all of this working, we will define some secondary terms and verify that some invariants of the translation hold. We will be call a sequent *translation sound* if this umbrella of conditions holds for the sequent.

2.3.1 Specification and soundness clauses

Let us introduce the different components of the translation and then define what it means for a sequent to be translation sound. The actual definitions of the components of the translation will be given in the next subsection.

Definition 2.8. For each symbol of $\mathscr{FOT}(\Sigma, \mathcal{T})$ we will associate the following data, all of it $\imath$-free by construction,

Sort-term t

We will to each sort-term t associate

- a sort-term $\mathtt{f}(t)$, the *unbinding* of t,
- a context $\mathtt{c}(t)$, the *freeing context* of t,
- and a proof-term $\mathtt{e}(t)$, the *existence witness* of t.

such that if $\Gamma \vdash t : \sigma$ is derivable and Γ is $\imath$-free then the following are derivable,

$$\Gamma \vdash \mathtt{c}(t)\ \mathtt{ctxt} \qquad \Gamma.\,\mathtt{c}(t) \vdash \mathtt{f}(t) : \sigma \qquad \Gamma \vdash \mathtt{e}(t) : \exists\,\mathtt{c}(t)\ \top$$

Formula φ

We will to each formula φ associate

- a formula $\mathtt{t}\,(\varphi)$, the *translation* of φ,
- proof-terms $\mathtt{i}(\varphi)$ and $\mathtt{j}(\varphi)$, the *translation equivalences* of φ,

such that if $\Gamma \vdash \varphi\ \mathtt{prop}$ is derivable and Γ is $\imath$-free then the following are derivable,

$$\Gamma \vdash \mathtt{t}\,(\varphi)\ \mathtt{prop} \qquad \Gamma \vdash \mathtt{i}(\varphi) : \mathtt{t}\,(\varphi) \to \varphi \qquad \Gamma \vdash \mathtt{j}(\varphi) : \varphi \to \mathtt{t}\,(\varphi)$$

and to each pair of formulas φ, ψ of the same kind

- proof-terms $\mathtt{k}\,(\varphi; \psi)$ and $\mathtt{h}\,(\varphi; \psi)$, the *equality equivalences* of φ, ψ,

such that over any context that judges φ and ψ equal we have

$$\Gamma \vdash \mathtt{k}\,(\varphi;\psi) : \mathtt{t}\,(\varphi) \to \mathtt{t}\,(\psi) \qquad \Gamma \vdash \mathtt{h}\,(\varphi;\psi) : \mathtt{t}\,(\psi) \to \mathtt{t}\,(\varphi)\,.$$

Proof-term ρ

We will to each proof-term ρ associate

- a proof-term $\mathtt{t}\,(\rho)$, the *translation* of ρ,

such that if $\Gamma \vdash \rho : \varphi$ is derivable then so is

$$\Gamma \vdash \mathtt{t}\,(\rho) : \mathtt{t}\,(\mathtt{p}(\rho))$$

where $\mathtt{p}(\rho)$ is the inferred type of ρ from Proposition 2.3.

Given the definitions in the next chapter, we take the following:

Definition 2.9. A sequent $\Gamma \vdash t : \sigma$ sort-term t is translation sound if the following sequents are derivable,

(i) $\Gamma \vdash \mathtt{c}(t)\ \mathtt{ctxt}$

(ii) $\Gamma.\,\mathtt{c}(t) \vdash \mathtt{f}(t) : \sigma$

(iii) $\Gamma \vdash \mathtt{e}(t) : \exists\,\mathtt{c}(t)\ \top$

(iv) $\Gamma.\,\mathtt{c}(t) \vdash t = \mathtt{f}(t) : \sigma$

and a derivable sequent $\Gamma \vdash t = \tau : \sigma$ with t, τ being sort-terms is translation sound if both $\Gamma \vdash t : \sigma$ and $\Gamma \vdash \tau : \sigma$ are translation sound.

Remark 2.10. It would be reasonable to include that if $\Gamma \vdash t = \tau : \sigma$ is derivable for t, τ sort-terms then so is $\Gamma.\,\mathtt{c}(t).\,\mathtt{c}(\tau) \vdash \mathtt{f}(t) = \mathtt{f}(\tau) : \sigma$ but that is satisfied by clause (iv).

Definition 2.11. A sequent $\Gamma \vdash \varphi\ \mathtt{prop}$ is translation sound if the following sequents are derivable,

(i) $\Gamma \vdash \mathtt{t}\,(\varphi)\ \mathtt{prop}$

(ii) $\Gamma.\,p : \mathtt{t}\,(\varphi) \vdash \mathtt{i}(\varphi)(p) : \varphi$

(iii) $\Gamma.\,p : \varphi \vdash \mathtt{j}(\varphi)(p) : \mathtt{t}\,(\varphi)$

and a sequent $\Gamma \vdash \varphi = \psi\ \mathtt{prop}$ is translation sound if the following are derivable,

(iv) $\Gamma \vdash \mathtt{k}\,(\varphi;\psi) : \mathtt{t}\,(\varphi) \to \mathtt{t}\,(\psi)$

(v) $\Gamma \vdash \mathtt{h}\,(\varphi;\psi) : \mathtt{t}\,(\psi) \to \mathtt{t}\,(\varphi)$

Remark 2.12. The above definition allows us to extend translation soundness to contexts over Γ, by first letting $\mathtt{t}\,(\sigma) = \sigma$ and $\mathtt{i}(\sigma), \mathtt{j}(\sigma)$ be identity substitutions. Then we let the following clauses extend the notions to a context,

$$\begin{aligned} \mathtt{t}(\Delta.\,x:\sigma) &= \mathtt{t}(\Delta).\,x:\sigma & \mathtt{t}(\Delta.\,p:\varphi) &= \mathtt{t}(\Delta).\,p:\mathtt{t}\,(\varphi\,[\mathtt{i}(\Delta)]) \\ \mathtt{i}(\Delta.\,x:\sigma) &= \mathtt{i}(\Delta).x & \mathtt{i}(\Delta.\,p:\varphi) &= \mathtt{i}(\Delta).\mathtt{i}(\varphi\,[\mathtt{i}(\Delta)])(p) \\ \mathtt{j}(\Delta.\,x:\sigma) &= \mathtt{j}(\Delta).x & \mathtt{j}(\Delta.\,p:\varphi) &= \mathtt{j}(\Delta).\mathtt{j}(\varphi\,[\mathtt{i}(\Delta)])(p) \end{aligned}$$

and we say that a sequent $\Gamma \vdash \Delta\ \mathtt{ctxt}$ is translation sound if

(i) $\Gamma \vdash \mathtt{t}(\Delta)\ \mathtt{ctxt}$

(ii) $\Gamma \vdash \mathtt{i}(\Delta) : \mathtt{t}(\Delta) \to \Delta$

(iii) $\Gamma \vdash \mathtt{j}(\Delta) : \Delta \to \mathtt{t}(\Delta)$

Definition 2.13. A sequent $\Gamma \vdash \rho : \varphi$ with proof-term ρ is said to be translation sound if $\Gamma \vdash \mathtt{p}(\rho) = \varphi\ \mathtt{prop}$ is translation sound and the following is derivable,

(i) $\Gamma \vdash \mathtt{t}\,(\rho) : \mathtt{t}\,(\mathtt{p}(\rho))$.

Definition 2.14. A rule of the type theory is *translation soundness preserving* if whenever the context Γ of the conclusion is $\imath$-free and all the assumption sequents

$$\Gamma.\,\Delta \vdash \mathcal{J}$$

have that all

$$\Gamma.\,\mathtt{t}(\Delta) \vdash \mathcal{J}\,[\mathtt{i}(\Delta)]$$

are translation sound then so is the conclusion.

2.3.2 Definitions and partial soundness results

Let us now give the concrete definitions. Note that we have chunked up the definitions into the categories of sort-terms, formulas and proof-terms. These definitions will actually depend on each other so the entire chapter is like a long definition. We will however intersperse the definitions with some results about translation soundness preservation because we will not require all the definitions laid out at the same time to prove them.

Definition 2.15. Define the *freeing context* $\mathtt{c}(t)$, the *unbinding* and the *existence witness* $\mathtt{e}(t)$ $\mathtt{f}(t)$ of a sort-term t in $\imath$-free context Γ by induction on its structure. First we make an auxiliary definition for how to iterate over a sequence $\vec{\tau}$ of terms,

$$\mathtt{f}(\vec{\tau}.t) = \mathtt{f}(\vec{\tau}).\,\mathtt{f}(t) \qquad\qquad \mathtt{c}(\vec{\tau}.t) = \mathtt{c}(\vec{\tau}).\,\mathtt{c}(t).$$

where we suppress some variable renaming from the notation to avoid variable collision. Now for the actual definition: The definite descriptions will give us our generating clause. We take

$$\begin{aligned} \mathtt{f}(\imath\, x : \sigma\ \varphi) &= x & \qquad \mathtt{c}(\imath\, x : \sigma\ \varphi) &= x : \sigma.\, p : \mathtt{t}\,(\varphi) \\ \mathtt{f}(f(\vec{\tau})) &= f(\mathtt{f}(\vec{\tau})) & \mathtt{c}(f(\vec{\tau})) &= \mathtt{c}(\vec{\tau}) \\ \mathtt{f}(x) &= x & \mathtt{c}(x) &= \Diamond \end{aligned}$$

Note that we left out the arguments for the definite description terms because they do not matter for the definition. Now for the existence witnesses. Once again the definite descriptions give us our generating clause,

$$\begin{aligned}\mathtt{e}(\imath\ x : \sigma\ \varphi(\varepsilon, \upsilon)) = &\exists\,\mathtt{E}\{\\ &\qquad x : \sigma \vdash \mathtt{t}\,(\varphi)\,,\\ &\qquad \exists\, x : \sigma\, \exists\, p : \mathtt{t}\,(\varphi)\ \top,\\ &\qquad \mathtt{k}\,(\mathtt{p}(\varepsilon); \exists\, x : \sigma\,\varphi)(\mathtt{t}\,(\varepsilon)),\\ &\qquad x : \sigma.\, p : \mathtt{t}\,(\varphi) \vdash \top\mathtt{I} : \top,\\ &\}.\end{aligned}$$

Now we turn to the auxiliary definition $\mathtt{e}(\vec{\tau})$ for how to iterate over a number of sort-terms,

$$\begin{aligned}\mathtt{e}(\vec{\tau}.\tau) = \exists\,\mathtt{E}\{\ &\mathtt{c}(\vec{\tau}) \vdash \top,\ \exists\,\mathtt{c}(\vec{\tau}.t)\ \top,\ \mathtt{e}(\vec{\tau}),\\ &\mathtt{v} : \mathtt{c}(\vec{\tau}).\_ : \top \vdash \exists\,\mathtt{E}\{\ \mathtt{c}(t) \vdash \top,\ \exists\,\mathtt{c}(\vec{\tau}.t)\ \top,\ \mathtt{e}(t),\\ &\qquad \mathtt{x} : \mathtt{c}(t).\_ : \top \vdash \exists\,\mathtt{I}(\ \mathtt{c}(\vec{\tau}.t) \vdash \top,\ \mathtt{v.x},\ \top\mathtt{I}\ )\ \}\ \}\end{aligned}$$

which allows us to give us our definitions for function symbols and variables,

$$\mathtt{e}(f(\vec{\tau})) = \mathtt{e}(\vec{\tau}), \qquad\qquad \mathtt{e}(x) = \top\mathtt{I}.$$

Note that the freeing context

$$\mathtt{c}(\imath\ x : \sigma\ \varphi) = x : \sigma.\, p : \mathtt{t}\,(\varphi)$$

calls the translation of the formula φ and likewise the existence witness $\mathtt{e}(\imath\ x : \sigma\ \varphi(\varepsilon, \upsilon))$ calls the translation of the proof-term ε.

As we will be eliminating on the existential witness a lot, we also use the following notation for brevity,

$$\begin{pmatrix}\mathtt{E}(t)\{\ \vartheta, & \\ \quad \mathtt{c}(t) \vdash \rho\ \}\end{pmatrix} = \begin{pmatrix}\exists\,\mathtt{E}\{\ \mathtt{c}(t) \vdash \top,\ \vartheta,\ \mathtt{e}(t), \\ \qquad\qquad \mathtt{c}(t).\_ : \top \vdash \rho\ \}\end{pmatrix}$$

Lemma 2.16. The symbol rules for the sort-terms are translation soundness preserving whenever the context Γ is $\imath$-free.

Proof. We handle the three syntactic cases (definite description term, function symbol, variable) separately.

Definite description sort-term

$$\frac{\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \varphi\ \mathtt{prop} \qquad \Gamma \vdash \varepsilon : \exists\, x : \sigma\ \varphi \qquad \Gamma.\, x : \sigma.\, p : \varphi.\, y : \sigma.\, q : \varphi\,[\,x \backslash y\,] \vdash \upsilon : x =_\sigma y}{\Gamma \vdash \imath\ x : \sigma\ \varphi(\varepsilon, \upsilon) : \sigma}$$

By translation soundness of the premises, the sequents

$$\Gamma.\, x : \sigma \vdash \mathtt{t}\,(\varphi)\ \ \mathtt{prop} \qquad \Gamma \vdash \mathtt{k}\,(\mathtt{p}(\varepsilon); \exists\, x : \sigma\ \varphi)(\mathtt{t}\,(\varepsilon)) : \exists\, x : \sigma\ \mathtt{t}\,(\varphi)$$

are derivable. The first one gives us that

$$\Gamma \vdash x : \sigma . \, p : \mathtt{t}\,(\varphi) \;\; \mathtt{ctxt} \qquad\qquad \Gamma . \, x : \sigma . \, p : \mathtt{t}\,(\varphi) \vdash x : \sigma$$

are derivable which gives us the correct typing for our freeing context and unbinding, respectively. We can also apply existential elimination to get typing for the existence witness,

$$\frac{\begin{array}{c}\Gamma \vdash \sigma \;\mathtt{sort} \qquad \Gamma . \, x : \sigma \vdash \mathtt{t}\,(\varphi) \;\; \mathtt{prop} \qquad \Gamma \vdash \exists\, x : \sigma \, \exists\, p : \mathtt{t}\,(\varphi) \;\; \top \;\mathtt{prop} \\ \Gamma \vdash \mathtt{k}\,(\mathtt{p}(\varepsilon); \exists\, x : \sigma \;\, \varphi)(\mathtt{t}\,(\varepsilon)) : \exists\, x : \sigma \;\, \mathtt{t}\,(\varphi) \\ \Gamma . \, x : \sigma . \, p : \mathtt{t}\,(\varphi) \vdash \exists\, \mathtt{I}(x, p, \top\mathtt{I}) : \exists\, x : \sigma \, \exists\, p : \mathtt{t}\,(\varphi) \;\; \top\end{array}}{\begin{array}{c}\Gamma \vdash \exists\, \mathtt{E}(\sigma, x.\mathtt{t}\,(\varphi)\,, \exists\, x : \sigma \, \exists\, p : \mathtt{t}\,(\varphi) \;\; \top, \mathtt{t}\,(\varepsilon)\,, x.p.\exists\, \mathtt{I}(x, p\top\mathtt{I})) \\ : \exists\, x : \sigma \, \exists\, p : \mathtt{t}\,(\varphi) \;\; \top\end{array}}$$

where all the premises are derivable by translation soundness of the premises. For the unbinding equality, with the notation that

$$t = \imath \; x : \sigma \;\; \varphi(\varepsilon, \upsilon) \qquad\qquad \rho = d \;\; x : \sigma \;\; \varphi(\varepsilon, \upsilon)$$

note that we have

$$\Gamma . \, x : \sigma . \, p : \mathtt{t}\,(\varphi) \vdash \upsilon(t, \rho, x, \mathtt{i}(\varphi)(p)) : t =_\sigma x$$

where derivability of

$$\Gamma . \, p : \mathtt{t}\,(\varphi) \vdash \mathtt{i}(\varphi)(p) : \varphi$$

is given by soundness induction hypothesis.

Function symbol

$$\frac{\Gamma \vdash \vec{\tau} : \vec{\sigma}}{\Gamma \vdash f(\vec{\tau}) : \sigma}$$

By translation soundness of the premises we get that

$$\Gamma . \, \mathtt{c}(\tau_i) \vdash \mathtt{f}(\tau_i) : \sigma_i$$

are derivable for every component $\tau_i : \sigma_i$ of $\vec{\tau} : \vec{\sigma}$. But by weakening we have that

$$\Gamma . \, \mathtt{c}(\vec{\tau}) \vdash \vec{\tau} : \vec{\sigma} \qquad\quad \text{and thus} \qquad\quad \Gamma . \, \mathtt{c}(\vec{\tau}) \vdash f(\mathtt{f}(\vec{\tau})) : \sigma$$

are derivable which means that we have correct typing of our unbinding. Similarly we can show correct typing of freeing context and unbinding equality. Correct typing of existence witness can be verified by iterating over $\vec{\tau}$.

Sort-variable

$$\frac{\vdash \Gamma.\, x : \sigma.\, \Delta \ \texttt{ctxt}}{\Gamma.\, x : \sigma.\, \Delta \vdash x : \sigma}$$

From this we directly get correct typing of the freeing context,

$$\Gamma.\, x : \sigma.\, \Delta \vdash \Diamond \ \texttt{ctxt}$$

correct typing of the unbinding,

$$\Gamma.\, x : \sigma.\, \Delta.\, \Diamond \vdash x : \sigma$$

and unbinding equality

$$\Gamma.\, x : \sigma.\, \Delta.\, \Diamond \vdash x = x : \sigma.$$

Also note that $\top\texttt{I}$ can be introduced from any context so we have correct typing of the existence witness.

□

Definition 2.17. Define the *translation* $\texttt{t}\,(\varphi)$ together with the *translation equivalences* $\texttt{i}(\varphi)$ and $\texttt{j}(\varphi)$ for formulas φ by induction on the structure of φ. For compound formulas the translation is trivial, combining using the given connective,

$$\begin{aligned} \texttt{t}\,(\varphi \vee \psi) &= \texttt{t}\,(\varphi) \vee \texttt{t}\,(\psi) \\ \texttt{t}\,(\exists\, x : \sigma\ \psi) &= \exists\, x : \sigma\ \texttt{t}\,(\psi) \\ \texttt{t}\,(\forall\, x : \sigma\ \psi) &= \forall\, x : \sigma\ \texttt{t}\,(\psi) \end{aligned}$$

and for the formulas with propositional quantifiers we also do a substitution along a translation equivalence,

$$\begin{aligned} \texttt{t}\,(\exists\, p : \varphi\ \psi) &= \exists\, p : \texttt{t}\,(\varphi)\ \ \texttt{t}\,(\psi\,[\texttt{i}(\varphi)]) \\ \texttt{t}\,(\forall\, p : \varphi\ \psi) &= \forall\, p : \texttt{t}\,(\varphi)\ \ \texttt{t}\,(\psi\,[\texttt{i}(\varphi)]) \end{aligned}$$

For atomic formulas we capture the freeing context of its arguments in an existential quantifier,

$$\texttt{t}\,(\Xi(\vec{\tau})) = \exists\, \texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau}))$$

Now we turn to the defining clauses for $\texttt{i}(\varphi)$ and $\texttt{j}(\varphi)$. For compound formulas, appeal to the equivalence of its subformulas. For instance, with $\varphi \vee \psi$ we take

$$\begin{aligned} \texttt{i}(\varphi \vee \psi)(\rho) = \vee\,\texttt{E}\{\ &\texttt{t}\,(\varphi)\,,\ \texttt{t}\,(\psi)\,,\ \varphi \vee \psi, \\ &p : \texttt{t}\,(\varphi) \vdash \vee\texttt{I}_L(\varphi, \psi, \texttt{i}(\varphi)(p)), \\ &q : \texttt{t}\,(\psi) \vdash \vee\texttt{I}_R(\varphi, \psi, \texttt{i}(\psi)(q)), \\ &\rho \\ \}. \end{aligned}$$

For $\exists p : \varphi\ \psi$, let $\psi\,[\,p \backslash \mathtt{i}(\varphi)(p')\,] = \psi'$ and take

$$
\begin{aligned}
\mathtt{i}(\exists p : \varphi\ \psi)(\rho) = \exists\,\mathtt{E}\{\ & p' : \mathtt{t}\,(\varphi) \vdash \mathtt{t}\,(\psi')\,, \\
& \exists p : \varphi\ \psi, \\
& \rho, \\
& p' : \mathtt{t}\,(\varphi)\,.\,q' : \mathtt{t}\,(\psi') \vdash \exists\,\mathtt{I}(\\
& \qquad p : \varphi \vdash \psi,\ \mathtt{i}(\varphi)(p'),\ \mathtt{i}(\psi')(q') \\
&) \\
\}.&
\end{aligned}
$$

For atomic formulas $\Xi(\vec{\tau})$ we take the translation

$$\mathtt{t}\,(\Xi(\vec{\tau})) = \exists\,\mathtt{c}(\vec{\tau})\ \Xi(\mathtt{f}(\vec{\tau}))$$

with translation equivalences given by

$$
\begin{aligned}
\mathtt{i}(\Xi(\vec{\tau}))(p) = \exists\,\mathtt{E}\{ & \\
& \mathtt{c}(\vec{\tau}) \vdash \Xi(\mathtt{f}(\vec{\tau})), \\
& \Xi(\vec{\tau}), \\
& p, \\
& \mathtt{c}(\vec{\tau}).\,q : \Xi(\mathtt{f}(\vec{\tau})) \vdash q, \\
\}&
\end{aligned}
$$

and

$$
\begin{aligned}
\mathtt{j}(\Xi(\vec{\tau}))(p) = \exists\,\mathtt{E}\{ & \\
& \mathtt{c}(\vec{\tau}) \vdash \top, \\
& \exists\,\mathtt{c}(\vec{\tau})\ \Xi(\mathtt{f}(\vec{\tau})), \\
& \mathtt{e}(\vec{\tau}), \\
& \mathtt{c}(\vec{\tau}).\,\_ : \top \vdash \exists\,\mathtt{I}(\ \mathtt{c}(\vec{\tau}) \vdash \Xi(\mathtt{f}(\vec{\tau})),\ \mathtt{f}(\vec{\tau}),\ p\ ), \\
\}.&
\end{aligned}
$$

Let us now turn to the definition of $\mathtt{k}\,(\varphi;\psi)$. We will only bother defining it when φ and ψ come from the same constructor as that is a prerequisite for $\Gamma \vdash \varphi = \psi\ \mathtt{prop}$, Lemma 2.4.

If φ, ψ are composite, we appeal to Proposition 2.2. If they are atomic formulas $\Xi(\vec{\tau}_1)$ and $\Xi(\vec{\tau}_2)$ we only eliminate on the existential witnesses,

$$
\begin{aligned}
\mathtt{k}\,(\Xi(\vec{\tau}_1);\Xi(\vec{\tau}_2))(p) = \mathtt{E}(\vec{\tau}_1)\{\ & \exists\,\mathtt{c}(\vec{\tau}_2)\ \Xi(\mathtt{f}(\vec{\tau}_2)), \\
& \mathtt{c}(\vec{\tau}_1) \vdash \mathtt{E}(\vec{\tau}_2)\{\ \exists\,\mathtt{c}(\vec{\tau}_2)\ \Xi(\mathtt{f}(\vec{\tau}_2)),\ p\ \}\ \}.
\end{aligned}
$$

and $\mathtt{h}\,(\Xi(\vec{\tau}_1);\Xi(\vec{\tau}_2))$ looks identical. (It does not matter which order we eliminate on the existential witnesses.)

Because the terms $\mathtt{k}\,(\mathtt{p}(\rho);\varphi)(\mathtt{t}\,(\rho))$ will be used heavily during the translation, we abbreviate that as simply

$$\mathtt{k}\,(\rho;\varphi).$$

Remark 2.18. Any composite formula $\bigcirc$ can be presented as

$$\bigcirc(\alpha_k, \alpha_{k(l)} \vdash \psi_l)$$

with symbol rule

$$\frac{\Gamma \vdash \alpha_k \ \mathtt{type} \qquad \Gamma.\, x_l : \alpha_{k(l)} \vdash \psi_l \ \mathtt{prop}}{\Gamma \vdash \bigcirc(\alpha_k, x_l : \alpha_{k(l)} \vdash \psi_l) \ \mathtt{prop}}$$

where indices k and l are understood to range over their values in the premise. With this presentation in mind, the translation according to the above definition is

$$\mathtt{t}\left(\bigcirc(\alpha_k, x_l : \alpha_{k(l)} \vdash \psi_l)\right) = \bigcirc(\mathtt{t}\,(\alpha_k)\,, x_l : \mathtt{t}\,(\alpha_k(l)) \vdash \mathtt{t}\,(\psi_l\,[\mathtt{i}(\alpha_k(l))]))$$

where we take $\mathtt{t}\,(\alpha) = \alpha$ and $\mathtt{i}(\alpha), \mathtt{j}(\alpha)$ to be the identity substitutions when α is a sort.

Lemma 2.19. The symbol rules for propositions are translation soundness preserving.

Proof. Let us treat the composite and atomic formulas separately.

Composite formulas

$$\frac{\Gamma \vdash \alpha_k \ \mathtt{type} \qquad \Gamma.\, x_l : \alpha_{k(l)} \vdash \psi_l \ \mathtt{prop}}{\Gamma \vdash \bigcirc(\alpha_k, x_l : \alpha_{k(l)} \vdash \psi_l) \ \mathtt{prop}}$$

By translation soundness of the first premise, we have that

$$\Gamma \vdash \mathtt{t}\,(\alpha_k) \ \mathtt{type} \qquad \Gamma \vdash \mathtt{i}(\alpha_{k(l)}) : \mathtt{t}\left(\alpha_{k(l)}\right) \to \alpha_{k(l)}$$

are derivable and therefore, by substitution, also that

$$\Gamma.\, x_l : \mathtt{t}\left(\alpha_{k(l)}\right) \vdash \psi_l\,[\mathtt{i}(\alpha_{k(l)})] \ \mathtt{prop}$$

is derivable. Translation soundness of the second premise gives us that

$$\Gamma.\, x_l : \mathtt{t}\left(\alpha_{k(l)}\right) \vdash \mathtt{t}\left(\psi_l\,[\mathtt{i}(\alpha_{k(l)})]\right) \ \mathtt{prop}$$

is derivable. Applying the symbol rule for the formula gives us correct typing of the translation

$$\frac{\Gamma \vdash \mathtt{t}\,(\alpha_k) \ \mathtt{type} \qquad \Gamma.\, x_l : \mathtt{t}\left(\alpha_{k(l)}\right) \vdash \mathtt{t}\left(\psi_l\,[\mathtt{i}(\alpha_{k(l)})]\right) \ \mathtt{prop}}{\bigcirc(\mathtt{t}\,(\alpha_k)\,, \mathtt{t}\left(\alpha_{k(l)}\right) \vdash \mathtt{t}\,(\psi_l\,[\mathtt{i}(\alpha_k(l))])) \ \mathtt{prop}}$$

The correct typing of the translation equivalences $\mathtt{i}, \mathtt{j}$ is given by appealing to the respective terms on subformulas in accordance with the fact that equivalent subformulas make up equivalent compound formulas in first order logic.

Atomic formulas

$$\frac{\Gamma \vdash \vec{\tau} : \vec{\sigma}}{\Gamma \vdash \Xi(\vec{\tau})\ \texttt{prop}}$$

In a similar way to correct typing of the unbinding of a function symbol, appealing to translation soundness of the premises and weakening gives us that

$$\Gamma.\,\texttt{c}(\vec{\tau}) \vdash \Xi(\texttt{f}(\vec{\tau}))\ \texttt{prop}$$

is derivable but then so is

$$\Gamma \vdash \exists\, \texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau}))\ \texttt{prop}$$

so we have correct typing of the translation. For correct typing of translation equivalences, let us first handle $\texttt{i}(\Xi(\vec{\tau}))$. The translation soundness of the premises gives us the unbinding equality,

$$\Gamma.\,\texttt{c}(\vec{\tau}) \vdash \vec{\tau} = \texttt{f}(\vec{\tau}) : \vec{\sigma} \quad \text{and therefore} \quad \Gamma.\,\texttt{c}(\vec{\tau}) \vdash \Xi(\vec{\tau}) = \Xi(\texttt{f}(\vec{\tau}))\ \texttt{prop}$$

are derivable. We can conclude that the following is derivable

$$\Gamma.\,\texttt{c}(\vec{\tau}).\,q : \Xi(\texttt{f}(\vec{\tau})) \vdash q : \Xi(\vec{\tau})$$

hence we can use existential elimination on p

$$\frac{\begin{array}{l}\Gamma.\,p : \texttt{t}\,(\Xi(\vec{\tau}))\,.\,\texttt{c}(\vec{\tau}) \vdash \Xi(\texttt{f}(\vec{\tau}))\ \texttt{prop} \\ \Gamma.\,p : \texttt{t}\,(\Xi(\vec{\tau})) \vdash \Xi(\vec{\tau})\ \texttt{prop} \qquad \Gamma.\,p : \texttt{t}\,(\Xi(\vec{\tau})) \vdash p : \exists\,\texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau})) \\ \Gamma.\,p : \texttt{t}\,(\Xi(\vec{\tau}))\,.\,\texttt{c}(\vec{\tau}).\,q : \Xi(\texttt{f}(\vec{\tau})) \vdash q : \Xi(\vec{\tau})\end{array}}{\Gamma.\,p : \texttt{t}\,(\Xi(\vec{\tau})) \vdash \exists\texttt{E}(\ldots) : \Xi(\vec{\tau})}$$

where the bottom term is precisely $\texttt{i}(\Xi(\vec{\tau}))$.

The procedure is similar for $\texttt{j}(\Xi(\vec{\tau}))$ in that we once again use the unbinding equality for $\vec{\tau}$ to show that

$$\Gamma.\,p : \Xi(\vec{\tau}).\,\texttt{c}(\vec{\tau}) \vdash p : \Xi(\texttt{f}(\vec{\tau}))$$

is derivable but this time we also utilize the existence witness of $\vec{\tau}$ in order to access the context $\texttt{c}(\vec{\tau})$ of the unbinding equality,

$$\frac{\begin{array}{l}\Gamma.\,p : \Xi(\vec{\tau}).\,\texttt{c}(\vec{\tau}) \vdash \top\ \texttt{prop} \\ \Gamma.\,p : \Xi(\vec{\tau}) \vdash \exists\,\texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau}))\ \texttt{prop} \qquad \Gamma.\,p : \Xi(\vec{\tau}) \vdash \texttt{e}(\vec{\tau}) : \exists\,\texttt{c}(\vec{\tau})\ \top \\ \Gamma.\,p : \Xi(\vec{\tau}).\,\texttt{c}(\vec{\tau}).\,\_ : \top \vdash \exists\texttt{I}(\ \texttt{c}(\vec{\tau}) \vdash \Xi(\texttt{f}(\vec{\tau})),\ \texttt{f}(\vec{\tau}),\ p\ ) : \exists\,\texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau}))\end{array}}{\Gamma.\,p : \Xi(\vec{\tau}) \vdash \exists\texttt{E}(\ldots) : \exists\,\texttt{c}(\vec{\tau})\ \Xi(\texttt{f}(\vec{\tau}))}$$

once again the bottom term is precisely $\texttt{j}(\Xi(\vec{\tau}))$.

□

We can similarly show that the equality rules for the propositions are soundness preserving. Let us highlight the most interesting case, for atomic formulas Ξ.

We will sloppily assume that $\vec{\tau}_1 = \vec{\tau}_2$. Let us bring up $\mathtt{k}\,(\Xi(\vec{\tau}_1); \Xi(\vec{\tau}_2))$,

$$\begin{aligned}\mathtt{k}\,(\Xi(\vec{\tau}_1); \Xi(\vec{\tau}_2))(p) = \mathtt{E}(\vec{\tau}_1)\{\ \exists\, \mathtt{c}(\vec{\tau}_2)\ \Xi(\mathtt{f}(\vec{\tau}_2)),& \\ \mathtt{c}(\vec{\tau}_1) \vdash \mathtt{E}(\vec{\tau}_2)\{\ \exists\, \mathtt{c}(\vec{\tau}_2)\ \Xi(\mathtt{f}(\vec{\tau}_2)),\ p\ \}\ \},&\end{aligned}$$

and we have that $\Gamma \vdash \vec{\tau}_1 : \vec{\sigma}$ and $\Gamma \vdash \vec{\tau}_2 : \vec{\sigma}$ are translation sound. We get by the unbinding equalities that over $\Gamma.\, \mathtt{c}(\vec{\tau}_1).\, \mathtt{c}(\vec{\tau}_2)$ we have

$$\Gamma.\, \mathtt{c}(\vec{\tau}_1).\, \mathtt{c}(\vec{\tau}_2) \vdash \Xi(\mathtt{f}(\vec{\tau}_1)) = \Xi(\mathtt{f}(\vec{\tau}_2))\ \mathtt{prop}$$

so that we indeed get

$$\Gamma.\, q : \Xi(\mathtt{f}(\vec{\tau}_1)).\, \mathtt{c}(\vec{\tau}_1).\, \mathtt{c}(\vec{\tau}_2) \vdash q : \Xi(\mathtt{f}(\vec{\tau}_2))$$

which means that, because the existence witnesses are derivable, we have

$$\Gamma.\, q : \Xi(\mathtt{f}(\vec{\tau}_1)) \vdash \mathtt{k}\,(\Xi(\mathtt{f}(\vec{\tau}_1)); \Xi(\mathtt{f}(\vec{\tau}_2))).$$

Lemma 2.20. If $\Gamma \vdash \varphi\ \mathtt{prop}$ is derivable and translation sound, we have

- $\Gamma \vdash \mathtt{k}\,(\varphi; \varphi) : \mathtt{t}\,(\varphi) \to \mathtt{t}\,(\varphi)$,
- $\Gamma \vdash \mathtt{h}\,(\varphi; \varphi) : \mathtt{t}\,(\varphi) \to \mathtt{t}\,(\varphi)$.

Proof. Induction on the structure of φ. □

Let us take stock of where we are so far. While the existential witnesses and the translation equivalences have called the translation of proof-terms in their definitions, the freeing context and unbinding of sort-terms, and the translation of propositions only call each other.

Their definitions are therefore finished at this point. Now we will prove some useful lemmas for how these syntactically interact with substitution, before resuming with a definition of the translation of the proof-terms.

Lemma 2.21. For any sort-term t, proposition φ, proof-variable p and proof-term ρ we have the following syntactic identities

$$\mathtt{c}(t)\,[\,p \backslash \rho\,] = \mathtt{c}(t\,[\,p \backslash \rho\,]) = \mathtt{c}(t) \qquad \mathtt{f}(t)\,[\,p \backslash \rho\,] = \mathtt{f}(t\,[\,p \backslash \rho\,]) = \mathtt{f}(t)$$

$$\mathtt{t}\,(\varphi)\ [\,p \backslash \rho\,] = \mathtt{t}\,(\varphi\,[\,p \backslash \rho\,]) = \mathtt{t}\,(\varphi)\,.$$

Proof. The first three are given by noting that they are all $\imath$-free and $\imath$-free sort-terms and propositions do not depend on proof-variables. The last identity follows essentially because unlike for formulas, the subexpressions of unbindings are never bound. □

Let us introduce a small substitution lemma to net us some relevant proof terms that will be used for defining the translation. The lemma depends on the grand substitution lemma that will be treated later.

Lemma 2.22. For any sort-term t, proposition φ, and sort-term a, we have that whenever $\Gamma \vdash a : \alpha$ and $\Gamma .\, x : \alpha$ are translation sound then we have

$$\Gamma . \vdash \mathtt{c}(t\,[\,x \backslash a\,])\ \mathtt{ctxt} \quad \text{and} \quad \Gamma .\, \mathtt{c}(a) \vdash \mathtt{c}(t)\,[\,x \backslash \mathtt{f}(a)\,]\ \mathtt{ctxt}$$

and

$$\Gamma \vdash \mathtt{t}\,(\varphi\,[\,x \backslash a\,])\ \mathtt{prop} \quad \text{and} \quad \Gamma .\, \mathtt{c}(a) \vdash \mathtt{t}\,(\varphi)\ [\,x \backslash \mathtt{f}(a)\,]\ \mathtt{prop}.$$

We also have terms $\mathtt{s}(\varphi; x \backslash a)$, $\mathtt{z}(\varphi; x \backslash a)$ such that we also have

$$\Gamma .\, \mathtt{c}(a).\, \Delta \vdash \mathtt{s}(\varphi; x \backslash a) : \mathtt{t}\,(\varphi)\ [\,x \backslash \mathtt{f}(a)\,] \to \mathtt{t}\,(\varphi\,[\,x \backslash a\,])$$

$$\Gamma .\, \mathtt{c}(a).\, \Delta \vdash \mathtt{z}(\varphi; x \backslash a) : \mathtt{t}\,(\varphi\,[\,x \backslash a\,]) \to \mathtt{t}\,(\varphi)\ [\,x \backslash \mathtt{f}(a)\,].$$

Proof. Apply the substitution lemma for sort-terms. We let

$$\begin{aligned}\mathtt{s}(\varphi; x \backslash a) &= \mathtt{i}(\varphi\,[\,x \backslash a\,]) \circ (\mathtt{j}(\varphi)\,[\,x \backslash \mathtt{f}(a)\,]) \\ &\qquad : \mathtt{t}\,(\varphi)\ [\,x \backslash \mathtt{f}(a)\,] \to \varphi\,[\,x \backslash \mathtt{f}(a)\,] = \varphi\,[\,x \backslash a\,] \to \mathtt{t}\,(\varphi\,[\,x \backslash a\,])\end{aligned}$$

where the middle equality holds by $a = \mathtt{f}(a)$ being derivable over $\mathtt{c}(a)$. Similarly,

$$\mathtt{z}(\varphi; x \backslash a) = \mathtt{j}(\varphi\,[\,x \backslash a\,]) \circ (\mathtt{i}(\varphi)\,[\,x \backslash \mathtt{f}(a)\,]) : \mathtt{t}\,(\varphi\,[\,x \backslash a\,]) \to \mathtt{t}\,(\varphi)\ [\,x \backslash \mathtt{f}(a)\,]$$

□

Most of the logical symbols can be handled in a uniform way, we will call them simple. We make this precise with the following definition.

Definition 2.23. The proof-terms for introduction rules which have the form

$$\frac{\Gamma \vdash \alpha_k\ \mathtt{type} \qquad \Gamma .\, x_l : \alpha_{k(l)} \vdash \psi_l\ \mathtt{prop} \qquad \Gamma .\, \Delta_m \vdash \rho_m : \varphi_m}{\Gamma \vdash \bigcirc\mathtt{I}(\alpha_k, \alpha_{k(l)} \vdash \psi_l, \Delta_m \vdash \rho_m) : \bigcirc(\alpha_k, \alpha_{k(l)} \vdash \psi_l)}$$

where the Δ_m only has types among α_k and ψ_l and likewise the φ_m are among α_k and ψ_l are called simple introduction proof-terms. Similarly, for elimination rules which have the form

$$\frac{\begin{array}{c}\Gamma \vdash \alpha_k\ \mathtt{type} \qquad \Gamma .\, \alpha_{k(l)} \vdash \psi_l\ \mathtt{prop} \\ \Gamma \vdash \delta : \bigcirc(\alpha_k, \alpha_{k(l)} \vdash \psi_l) \qquad \Gamma .\, \Delta_m \vdash \rho_m : \varphi_m\end{array}}{\Gamma \vdash \bigcirc\mathtt{E}(\alpha_k, \alpha_{k(l)} \vdash \psi_l, \delta, \Delta_m \vdash \rho_m) : \alpha_0}$$

the proof-term is called a simple elimination proof-term. The simple proof-terms are truth intro, false elim, disjunction intros and elim, existential elims and universal intros.

Definition 2.24. Define the translation $\mathtt{t}\,(\rho)$ of the proof-term ρ by induction on its structure. First for the main generating clause, the proof-term for definite descriptions. Note that $\mathtt{t}\,(\varphi)$ and $\mathtt{t}\,(\varphi)\ [x \backslash \imath\, x : \sigma\ \varphi(\varepsilon, \upsilon)]$ are syntactically identical. With Lemma 2.22 we take the translation to be

$$
\begin{aligned}
\mathtt{t}\,(\text{ð}\ x : \sigma\ \varphi(\varepsilon,\upsilon)) = \exists\,\mathtt{E}\{ &\\
&x : \sigma \vdash \mathtt{t}\,(\varphi)\,,\\
&\mathtt{t}\,(\varphi\,[x \backslash \imath\, x : \sigma\ \varphi(\varepsilon,\upsilon)\,])\,,\\
&\mathtt{k}\,(\mathtt{p}(\varepsilon); \exists\, x : \sigma\ \varphi)\{\mathtt{t}\,(\varepsilon)\},\\
&x : \sigma . \, p : \mathtt{t}\,(\varphi) \vdash \mathtt{s}(\varphi; x \backslash \imath\, x : \sigma\ \varphi(\varepsilon,\upsilon))(p)\\
\}.&
\end{aligned}
$$

For translating simple proof-terms ρ, the main tool in constructing $\mathtt{t}\,(\rho)$ is by looking at what typing $\mathtt{t}\,(\varphi)$ the term ought to have, and then replacing ρ by $\mathtt{k}\,(\rho;\varphi)$. We first need to do substitution with $\mathtt{i}(\varphi)$ whenever a term depends on a proof-term variable. For instance, for disjunction elimination we take

$$
\begin{aligned}
\mathtt{t}\,(\vee\mathtt{E}(\varphi,\psi,\vartheta,\pi,p.\rho,q.\delta)) = \vee\,\mathtt{E}\{\ &\mathtt{t}\,(\varphi)\,,\ \mathtt{t}\,(\psi)\,,\ \mathtt{t}\,(\vartheta)\,,\\
&\mathtt{k}\,(\pi; \varphi \vee \psi),\\
&p : \mathtt{t}\,(\varphi) \vdash \mathtt{k}\,(\rho\,[\mathtt{i}(\varphi)]; \vartheta),\\
&q : \mathtt{t}\,(\psi) \vdash \mathtt{k}\,(\delta\,[\mathtt{i}(\psi)]; \vartheta),\\
\}.&
\end{aligned}
$$

For prop-existential intro and prop-universal elim are similar because $\mathtt{t}\,(\psi)$ does not depend on proof-term variables and therefore

$$\mathtt{t}\,(\psi\,[p \backslash \rho\,]) = \mathtt{t}\,(\psi)\ [p \backslash \rho\,] = \mathtt{t}\,(\psi)$$

so we can translate the entire expression by simply translating the subexpressions in the same way we did here. For sort-existential intro and sort-universal elim, do a similar kind of translation but wrap the entire expression in an existential elimination on $\mathtt{e}(t)$ and apply Lemma 2.22 where necessary to handle substitutions. For instance, with sort-universal elim we take

$$
\begin{aligned}
\mathtt{t}\,(\forall\mathtt{E}\{x : \sigma \vdash \psi, \rho, t\}) = \mathtt{E}(t)\{\ &\mathtt{t}\,(\psi\,[x \backslash t\,])\,,\\
&\mathtt{c}(t) \vdash \mathtt{z}(\psi; x \backslash t)(\ \forall\mathtt{E}\{\ x : \sigma \vdash \mathtt{t}\,(\psi)\,,\ \mathtt{k}\,(\rho;\psi),\ \mathtt{f}(t),\ \}\ )\ \}.
\end{aligned}
$$

For equality intro we only need wrap the expression in an existential elimination on $\mathtt{e}(t)$ and translate subexpressions,

$$
\begin{aligned}
\mathtt{t}\,(\,{=}\mathtt{I}(\sigma,t)) = \mathtt{E}(t)\{\ &\exists\,\mathtt{c}(t)\ \mathtt{f}(t) =_\sigma \mathtt{f}(t),\\
&\mathtt{w} : \mathtt{c}(t) \vdash \exists\,\mathtt{I}(\ \mathtt{c}(t) \vdash \mathtt{f}(t) =_\sigma \mathtt{f}(t),\ \mathtt{w},\ {=}\mathtt{I}(\sigma, \mathtt{f}(t))\,)\ \}.
\end{aligned}
$$

For variables the translation acts as the identity. For axiom terms $\mathcal{A}$, we simply take

$$\mathtt{t}\,(\mathcal{A}(\rho_1,\ldots,\rho_n)) = \mathcal{A}(\mathtt{k}\,(\rho_1;\theta_1),\ldots,\mathtt{k}\,(\rho_n;\theta_n))\}$$

Now we have finished defining all the components of the translation.

Lemma 2.25. The symbol rules for proof-terms are translation soundness preserving.

Proof. In all cases, the translated term will be typed as the translation of the inferred type, which follows because we apply the appropriate symbol rule. For the equality equivalences, note that what we will want to derive will be

$$\mathtt{k}\,(\mathtt{p}(\rho);\mathtt{p}(\rho))$$

which we have by Lemma 2.20.

Definite description proof-term

$$\frac{\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\, x : \sigma \vdash \varphi\ \mathtt{prop} \\ \Gamma \vdash \varepsilon : \exists\, x : \sigma\ \varphi \qquad \Gamma.\, x : \sigma.\, p : \varphi.\, y : \sigma.\, q : \varphi\,[\, x \backslash y\,] \vdash \upsilon : x =_\sigma y}{\Gamma \vdash \partial\ x : \sigma\ \varphi(\varepsilon, \upsilon) : \psi\,[\, x \backslash \imath\ x : \sigma\ \varphi(\varepsilon, \upsilon)\,]}$$

Translation soundness of premises gives us that

$$\Gamma.\, x : \sigma \vdash \mathtt{t}\,(\varphi)\ \mathtt{prop} \qquad \Gamma \vdash \mathtt{k}\,(\varepsilon; \exists\, x : \sigma\ \varphi) : \exists\, x : \sigma\ \mathtt{t}\,(\varphi)$$

are derivable, and by applying Lemma 2.22 we get that

$$\Gamma.\, x : \sigma.\, p : \mathtt{t}\,(\varphi)\,.\, q : \mathtt{t}\,(\varphi)\ [\, x \backslash \mathtt{f}(\imath\ x : \sigma\ \varphi)\,] \vdash \\ \mathtt{s}(\varphi; x \backslash \imath\ x : \sigma\ \varphi)(q) : \mathtt{t}\,(\varphi\,[\, x \backslash \imath\ x : \sigma\ \varphi\,])$$

is derivable but since $\mathtt{f}(\imath\ x : \sigma\ \varphi)$ is x we have

$$\mathtt{t}\,(\varphi)\ [\, x \backslash \mathtt{f}(\imath\ x : \sigma\ \varphi)\,] = \mathtt{t}\,(\varphi)$$

and we can use substitution to derive

$$\Gamma.\, x : \sigma.\, p : \mathtt{t}\,(\varphi) \vdash \mathtt{s}(\varphi; x \backslash \imath\ x : \sigma\ \varphi)(p) : \mathtt{t}\,(\varphi\,[\, x \backslash \imath\ x : \sigma\ \varphi\,])\,.$$

Now we can apply existential elimination

$$\frac{\Gamma.\, x : \sigma \vdash \mathtt{t}\,(\varphi)\ \mathtt{prop} \\ \Gamma \vdash \mathtt{t}\,(\varphi\,[\, x \backslash \imath\ x : \sigma\ \varphi\,])\ \mathtt{prop} \qquad \Gamma \vdash \mathtt{k}\,(\varepsilon; \exists\, x : \sigma\ \varphi) : \exists\, x : \sigma\ \mathtt{t}\,(\varphi) \\ \Gamma.\, x : \sigma.\, p : \mathtt{t}\,(\varphi) \vdash \mathtt{s}(\varphi; x \backslash \imath\ x : \sigma\ \varphi)(p) : \mathtt{t}\,(\varphi\,[\, x \backslash \imath\ x : \sigma\ \varphi\,])}{\Gamma \vdash \exists\mathtt{E}(\ldots) : \mathtt{t}\,(\varphi\,[\, x \backslash \imath\ x : \sigma\ \varphi\,])}$$

where the bottom term is our translation and the type is the translation of the inferred type for the definite description proof-term.

Simple proof-terms

Let us illustrate with prop-universal introduction.

$$\frac{\Gamma \vdash \varphi\ \mathtt{prop} \qquad \Gamma.\, p : \varphi \vdash \psi\ \mathtt{prop} \qquad \Gamma.\, p : \varphi \vdash \rho : \psi}{\Gamma \vdash \forall\mathtt{I}(\ p : \varphi \vdash \psi,\ p : \varphi \vdash \rho\ ) : \forall\, p : \varphi\ \psi}$$

By translation soundness of the premises, we get that

$$\Gamma \vdash \mathtt{t}\,(\varphi)\ \ \mathtt{prop} \qquad\qquad \Gamma.\,p : \mathtt{t}\,(\varphi) \vdash \mathtt{t}\,(\psi)\ \ \mathtt{prop}$$

$$\Gamma.\,p : \mathtt{t}\,(\varphi) \vdash \mathtt{k}\,(\rho\,[\mathtt{i}(\varphi)];\psi) : \mathtt{t}\,(\varphi)$$

are derivable, so we apply the symbol rule and get

$$\Gamma \vdash \forall\mathtt{I}\{\ p : \mathtt{t}\,(\varphi) \vdash \mathtt{t}\,(\psi)\,,\ p : \mathtt{t}\,(\varphi)) \vdash \mathtt{k}\,(\rho\,[\mathtt{i}(\varphi)];\psi)\} \\ : \forall\,p : \mathtt{t}\,(\varphi)\ \ \mathtt{t}\,(\psi)\,.$$

This is just what we wanted, we have that the translation of our term is typed as the translation of its inferred type $\mathtt{t}\,(\forall\,p : \varphi\ \psi)$.

Prop-existential intro, prop-universal elim

Let us illustrate with prop-existential intro.

$$\frac{\Gamma \vdash \varphi\ \mathtt{prop} \qquad \Gamma.\,p : \varphi \vdash \psi\ \mathtt{prop} \qquad \Gamma \vdash \rho : \varphi \qquad \Gamma \vdash \delta : \psi\,[\,p\backslash\rho\,]}{\Gamma \vdash \exists\,\mathtt{I}(\ldots) : \exists\,p : \varphi\ \psi}$$

Translation soundness of the premises gives us exactly what we need to apply the symbol rule and derive what we want. Let us just highlight what happens with the last premise. Translation soundness gives us

$$\Gamma \vdash \mathtt{k}\,(\delta;\varphi\,[\,p\backslash\rho\,]) : \mathtt{t}\,(\varphi\,[\,p\backslash\rho\,])$$

but as $\mathtt{t}\,(\varphi\,[\,p\backslash\rho\,]) = \mathtt{t}\,(\varphi)$ this is what we need in order to apply the symbol rule and get the translated term.

Sort-existential intro, sort-universal elim

Let us illustrate with sort-universal elim.

$$\frac{\Gamma \vdash \sigma\ \mathtt{sort} \qquad \Gamma.\,x : \sigma \vdash \varphi\ \mathtt{prop} \qquad \Gamma \vdash \pi : \forall\,x : \sigma\ \varphi \qquad \Gamma \vdash t : \sigma}{\Gamma \vdash \forall\mathtt{E}(\,x : \sigma \vdash \varphi,\ \pi,\ t\,) : \varphi\,[\,x\backslash t\,]}$$

By translation soundness of t we are allowed to eliminate on its existential witness. Furthermore we can get

$$\Gamma.\,\mathtt{c}(t) \vdash \forall\mathtt{E}(\,x : \sigma \vdash \mathtt{t}\,(\varphi)\,,\,\mathtt{k}\,(\pi;\forall\,x : \sigma\ \varphi),\,\mathtt{f}(t)) : \mathtt{t}\,(\varphi)\ [\,x\backslash\mathtt{f}(t)\,]$$

by translation soundness of $\pi : \forall\,x : \sigma\ \varphi$ and φ. Wrapping this in $\mathtt{z}(\varphi;x\backslash t)$ and an elimination on the existential witness gives us what we wanted.

Variable

$$\frac{\vdash \Gamma.\,p : \varphi.\,\Delta\ \mathtt{ctxt}}{\Gamma.\,p : \varphi.\,\Delta \vdash p : \varphi}$$

Note that φ is $\imath$-free because the bottom context is $\imath$-free.

We will not show more cases. □

2.3.3 Substitution and soundness

We will soon show that all $\imath$-free sequents are translation sound. First we show that the substitution rules are soundness preserving.

Lemma 2.26. The substitution rules are translation soundness preserving.

Proof. Let us split into two cases, either we are substituting in a proof-term or we are substituting in a sort-term. Let us first deal with the sort-term case,

$$\frac{\Gamma \vdash a : \alpha \qquad \Gamma .\, x : \alpha .\, \Delta \vdash \mathcal{J}}{\Gamma .\, \Delta\, [\, x \backslash a\,] \vdash \mathcal{J}\, [\, x \backslash a\,]}$$

$\imath$-freeness of $\Delta\, [\, x \backslash a\,]$ gives us that Δ is $\imath$-free just as the previous case, and also that $\Delta\, [\, x \backslash a\,] = \Delta\, [\, x \backslash \mathtt{f}(a)\,]$. Either way, if we translate the assumptions and then apply the substitution, we get

$$\Gamma .\, \Delta\, [\, x \backslash a\,] .\, \mathtt{c}(a) \vdash \mathcal{J}\, [\, x \backslash \mathtt{f}(a)\,]$$

where we could swap the order of $\Delta\, [\, x \backslash a\,]$ and $\mathtt{f}(a)$ because they are independent over Γ. We will now proceed by induction on the structure of the argument that if the assumptions

$$\Gamma \vdash a : \alpha \quad \text{and} \quad \Gamma .\, x : \alpha .\, \Delta \vdash \mathcal{J}$$

are translation sound, then so is the sequent

$$\Gamma .\, \Delta\, [\, x \backslash a\,] \vdash \mathcal{J}\, [\, x \backslash a\,].$$

Let us proceed, starting with the sort-terms.

Sort-variable

Let t be y. We need to split the cases on whether $x = y$ or $x \neq y$.

When $x = y$, the context $\mathtt{c}(a) .\, \mathtt{c}(y)\, [\, x \backslash \mathtt{f}(a)\,]$ (which we can derive over $\Gamma .\, \Delta\, [\, x \backslash a\,]$ by substitution) is actually identical with $\mathtt{c}(y\, [\, x \backslash a\,])$ so we get the freeing context. We also get that $\mathtt{f}(x)\, [\, x \backslash \mathtt{f}(a)\,]$ is identical with $\mathtt{f}(y\, [\, x \backslash a\,])$, the unbinding. By translation soundness of a, we certainly have

$$\Gamma .\, \mathtt{c}(a) \vdash \top\mathtt{I} : \top \quad \text{and} \quad \Gamma \vdash \mathtt{e}(a) : \exists\, \mathtt{c}(a)\ \top$$

which gives us our existence witness. In a similar fashion, we get by translation soundness for a that both the unbinding equality is derivable.

If $x \neq y$, of course $\lozenge$ is derivable, and as

$$\mathtt{f}(y\, [\, x \backslash a\,]) = y$$

we have that both the unbinding is over the freeing context. Also, the existence witness is quite simply derivable. The unbinding equality are also derivable, quite tautological.

Definite description sort-term

We have

$$\Gamma.\, x : \alpha.\, \Delta \vdash \imath\; y : \sigma\; \varphi(\varepsilon, \upsilon).$$

Therefore we in particular also have

$$\Gamma.\, x : \alpha.\, \Delta y : \sigma \vdash \varphi \; \texttt{prop} \qquad\qquad \Gamma.\, x : \alpha.\, \Delta \vdash \varepsilon : \exists\, y : \sigma\; \varphi$$

which means by inductive hypothesis that the are derivable,

$$\Gamma.\, \Delta\,[\,x\backslash a\,].\, y : \sigma \vdash \texttt{t}\,(\varphi\,[\,x\backslash a\,])\;\texttt{prop} \quad \text{and}$$

$$\Gamma \vdash \texttt{t}\,(\varepsilon\,[\,x\backslash a\,]) : \exists\, y : \sigma\; \texttt{t}\,(\varphi\,[\,x\backslash a\,])$$

which nets us our freeing context and existence witness. We get the unbinding equality by applying $\texttt{i}$ to $p : \texttt{t}\,(\varphi\,[\,x\backslash a\,])$, allows us to use $\upsilon\,[\,x\backslash a\,]$ to get propositional equality from which we can get the judgemental equality.

Function symbol

As $\texttt{c}(f(\vec{\tau})\,[\,x\backslash a\,])$ is the concatenation of $\texttt{c}(\tau_i\,[\,x\backslash a\,])$ we get that the freeing context is derivable by inductive hypothesis.

Now because the unbindings $\texttt{f}(\tau_i\,[\,x\backslash a\,])$ are derivable terms over $\Gamma.\, \Delta\,[\,x\backslash a\,].\, \texttt{c}(\tau_i\,[\,x\backslash a\,])$ we get the derivability of the unbinding of $f(\vec{\tau}\,[\,x\backslash a\,])$ by applying weakening to the terms.

Now we will treat the formulas.

Atomic proposition

The derivability of

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \exists\, \texttt{c}(\vec{\tau}\,[\,x\backslash a\,])\; \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,]))\;\texttt{prop}$$

follows from the inductive hypothesis on $\vec{\tau}$. For the translation equivalences, we need to be a bit craftier. Let us inspect the term what we need to derive for $\texttt{j}$,

$$\begin{aligned}
&\Gamma.\, \Delta\,[\,x\backslash a\,].\, q : \Xi(\vec{\tau}\,[\,x\backslash a\,]) \vdash \exists\,\texttt{E}\{\\
&\qquad \texttt{c}(\vec{\tau}\,[\,x\backslash a\,]) \vdash \top,\ \exists\, \texttt{c}(\vec{\tau}\,[\,x\backslash a\,])\; \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,])),\ \texttt{e}(\vec{\tau}\,[\,x\backslash a\,]),\\
&\quad \texttt{z} : \texttt{c}(\vec{\tau}\,[\,x\backslash a\,]) \vdash \exists\,\texttt{I}(\ \texttt{c}(\vec{\tau}\,[\,x\backslash a\,]).\, _ : \top \vdash \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,])),\ \texttt{z},\ q,\ )\ \}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad : \exists\, \texttt{c}(\vec{\tau}\,[\,x\backslash a\,])\; \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,])).
\end{aligned}$$

We almost get derivability directly from the derivability of the subterms. The only thing we need to prove as well is that

$$q : \Xi(\vec{\tau}\,[\,x\backslash a\,]).\, \texttt{z} : \texttt{c}(\vec{\tau}\,[\,x\backslash a\,]) \vdash q : \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,]))\,[\texttt{z}]$$

but here we apply the unbinding equality to get

$$\Xi(\vec{\tau}\,[\,x\backslash a\,]) = \Xi(\texttt{f}(\vec{\tau}\,[\,x\backslash a\,]))$$

from which it follows.

Composite formulas

Let us illustrate with $\exists p : \varphi\ \psi$. The rest of the cases work the same way. We have

$$\Gamma.\, x : \alpha.\, \Delta \vdash \exists p : \varphi\ \psi\ \texttt{prop}$$

which gives us

$$\Gamma.\, x : \alpha.\, \Delta \vdash \varphi\ \texttt{prop} \quad \text{and} \quad \Gamma.\, x : \alpha.\, \Delta.\, p : \varphi \vdash \psi\ \texttt{prop}.$$

We apply inductive hypothesis that the following are derivable

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \texttt{t}\,(\varphi\,[\,x\backslash a\,])\ \texttt{prop}$$

$$\Gamma.\, \Delta\,[\,x\backslash a\,].\, p : \texttt{t}\,(\varphi\,[\,x\backslash a\,]) \vdash \texttt{t}\,(\psi\,[\,x\backslash a\,])\ \texttt{prop}.$$

(Here we used that $\texttt{t}\,(\psi\,[\,x\backslash a\,]\,[\texttt{i}(\varphi\,[\,x\backslash a\,])]) = \texttt{t}\,(\psi\,[\,x\backslash a\,])$.) This gives us that the following is derivable,

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \exists p : \texttt{t}\,(\varphi\,[\,x\backslash a\,])\ \ \texttt{t}\,(\psi\,[\,x\backslash a\,])\,.$$

For the translation equivalences, note that as $\texttt{i}\langle\exists\rangle$ and $\texttt{j}\langle\exists\rangle$ of Proposition 2.2 are defined by the translation equivalences for the subexpressions so we can apply the inductive hypothesis.

Simple proof-terms, prop-universal elim, prop-existential intro

Let us treat the case where our proof-term is prop-existential elimination for an illustrating example. Here's what the term δ looks like when it's derivable in a context,

$$\Gamma.\, x : \alpha.\, \Delta \vdash \exists\texttt{E}\{\ p : \varphi \vdash \psi,\ \theta,\ \pi,\ p : \varphi.\, q : \psi \vdash \rho\ \} : \vartheta$$

and the arguments look as follows after we substitute in a,

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \varphi\,[\,x\backslash a\,]\ \texttt{prop} \qquad\qquad \Gamma.\, \Delta\,[\,x\backslash a\,].\, p : \varphi \vdash \psi\ \texttt{prop}$$

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \pi : \exists \varphi\,[\,x\backslash a\,]\ \psi\,[\,x\backslash a\,]$$

$$\Gamma.\, p : \varphi\,[\,x\backslash a\,].\, q : \psi\,[\,x\backslash a\,] \vdash \rho\,[\,x\backslash a\,] : \theta\,[\,x\backslash a\,]$$

If we apply existential elimination we can therefore derive

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \delta\,[\,x\backslash a\,] : \theta\,[\,x\backslash a\,]$$

on the other hand direct application of substitution gives us

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \delta\,[\,x\backslash a\,] : \vartheta\,[\,x\backslash a\,]$$

and therefore by Proposition 2.5 they are judged equal,

$$\Gamma.\, \Delta\,[\,x\backslash a\,] \vdash \theta\,[\,x\backslash a\,] = \vartheta\,[\,x\backslash a\,]\ \texttt{prop}.$$

Apply the inductive hypothesis to this judgement to get $\mathtt{k}, \mathtt{h}$ between $\mathtt{t}\,(\theta\,[\,x\backslash a\,])$ and $\mathtt{t}\,(\vartheta\,[\,x\backslash a\,])$. If we apply the inductive hypothesis to the arguments we can follow up with the existential elimination rule to get

$$\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \mathtt{t}\,(\delta\,[\,x\backslash a\,]) : \mathtt{t}\,(\theta\,[\,x\backslash a\,])$$

and note that because $\theta\,[\,x\backslash a\,]$ is $\mathtt{p}(\rho\,[\,x\backslash a\,])$ this gives us translation soundness for this case, as we have

(i) $\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \mathtt{t}\,(\delta\,[\,x\backslash a\,]) : \mathtt{t}\,(\mathtt{p}(\delta\,[\,x\backslash a\,]))\,,$

(ii) $\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \mathtt{k}\,(\mathtt{p}(\delta\,[\,x\backslash a\,]); \vartheta\,[\,x\backslash a\,]) : \mathtt{t}\,(\mathtt{p}(\delta\,[\,x\backslash a\,])) \to \mathtt{t}\,(\vartheta\,[\,x\backslash a\,])\,,$

(iii) $\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \mathtt{h}\,(\mathtt{p}(\delta\,[\,x\backslash a\,]); \vartheta\,[\,x\backslash a\,]) : \mathtt{t}\,(\vartheta\,[\,x\backslash a\,]) \to \mathtt{t}\,(\mathtt{p}(\delta\,[\,x\backslash a\,])).$

Let us also treat left disjunction intro, as the intro rules work slightly differently.

$$\Gamma.\,x : \alpha.\,\Delta \vdash \vee\mathtt{I}_L(\,\varphi_1,\,\varphi_2,\,\rho\,) : \vartheta$$

The difference is that we do not have $\mathtt{p}(\delta)$ directly as an argument of δ (as it is an introduction rule) so we need to work a bit more for our equality equivalences $\mathtt{k}, \mathtt{h}$. By admissibility and Propositions 2.4, 2.5 we have that $\vartheta = \psi_1 \vee \psi_2$ for some ψ_1, ψ_2. We also have

$$\Gamma.\,x : \alpha.\,\Delta \vdash \varphi_1 = \psi_1\ \mathtt{prop} \quad \text{and} \quad \Gamma.\,x : \alpha.\,\Delta \vdash \varphi_2 = \psi_2\ \mathtt{prop}.$$

Inductive hypothesis applied to the judgements

$$\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \varphi_1\,[\,x\backslash a\,] = \psi_1\,[\,x\backslash a\,]\ \mathtt{prop}$$

$$\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \varphi_2\,[\,x\backslash a\,] = \psi_2\,[\,x\backslash a\,]\ \mathtt{prop}$$

nets us our $\mathtt{h}$, $\mathtt{k}$ for the pairs φ_1, ψ_1 and φ_2, ψ_2. But this gives us our $\mathtt{h}, \mathtt{k}$ as they are defined via $\mathtt{i}\langle\vee\rangle, \mathtt{j}\langle\vee\rangle$ of Proposition 2.2 on

$$\mathtt{k}\,(\varphi_1\,[\,x\backslash a\,]; \psi_1\,[\,x\backslash a\,]) \qquad \mathtt{h}\,(\varphi_1\,[\,x\backslash a\,]; \psi_1\,[\,x\backslash a\,])$$

$$\mathtt{k}\,(\varphi_2\,[\,x\backslash a\,]; \psi_2\,[\,x\backslash a\,]) \qquad \mathtt{h}\,(\varphi_2\,[\,x\backslash a\,]; \psi_2\,[\,x\backslash a\,]).$$

Sort-universal elim, sort-existential intro

This one is a bit different because we need to apply the inductive hypothesis twice. Let us display sort-existential intro.

$$\Gamma.\,x : \alpha.\,\Delta \vdash \exists\,\mathtt{I}(\,y : \sigma \vdash \varphi,\,t,\,\rho\,) : \vartheta$$

Just as in the left intro for disjunction we can get that $\vartheta = \exists\,y : \sigma\ \psi$ for some ψ such that

$$\Gamma.\,x : \alpha.\,\Delta.\,y : \sigma \vdash \varphi = \psi\ \mathtt{prop}$$

and we use this to get

$$\begin{aligned}\Gamma.\,\Delta\,[\,x\backslash a\,] &\vdash \mathtt{k}\,(\exists\, y:\sigma\ \varphi\,[\,x\backslash a\,]; \exists\, y:\sigma\ \psi\,[\,x\backslash a\,])\\ &\qquad : \exists\, y:\sigma\ \mathtt{t}\,(\varphi\,[\,x\backslash a\,]) \to \exists\, y:\sigma\ \mathtt{t}\,(\psi\,[\,x\backslash a\,])\ \text{ and}\\ \Gamma.\,\Delta\,[\,x\backslash a\,] &\vdash \mathtt{h}\,(\exists\, y:\sigma\ \varphi\,[\,x\backslash a\,]; \exists\, y:\sigma\ \psi\,[\,x\backslash a\,])\\ &\qquad : \exists\, y:\sigma\ \mathtt{t}\,(\psi\,[\,x\backslash a\,]) \to \exists\, y:\sigma\ \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\,.\end{aligned}$$

We can also first do a substitution of x for a on φ, where translation soundness is preserved by inductive hypothesis and we get

$$\Gamma.\,\Delta\,[\,x\backslash a\,].\,y:\sigma \vdash \mathtt{j}(\varphi\,[\,x\backslash a\,]) : \varphi\,[\,x\backslash a\,] \to \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\,.$$

If we directly apply substitution of y for $\mathtt{f}(t)$ to $\mathtt{j}(\varphi\,[x\backslash a])$ over the context $\mathtt{c}(t)$ we get

$$\begin{aligned}\Gamma.\,\Delta\,[\,x\backslash a\,].\,\mathtt{c}(t) &\vdash \mathtt{j}(\varphi\,[\,x\backslash a\,])\,[\,y\backslash \mathtt{f}(t)\,]\\ &\qquad : \varphi\,[\,x\backslash a\,]\,[\,y\backslash \mathtt{f}(t)\,] \to \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\ [\,y\backslash \mathtt{f}(t)\,]\end{aligned}$$

whereas if we apply the inductive hypothesis for substituting y for t in $\mathtt{t}\,(\varphi\,[\,x\backslash a\,])$ we get

$$\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash \mathtt{i}(\varphi\,[\,x\backslash a\,]\,[\,y\backslash t\,]) : \mathtt{t}\,(\varphi\,[\,x\backslash a\,]\,[\,y\backslash t\,]) \to \varphi\,[\,x\backslash a\,]\,[\,y\backslash t\,].$$

These can be composed over $\mathtt{c}(t)$ (by unbinding equality) to get

$$\begin{aligned}\Gamma.\,\Delta\,[\,x\backslash a\,].\,\mathtt{c}(t) &\vdash \mathtt{s}(\varphi\,[\,x\backslash a\,]; y\backslash t)\\ &\qquad : \mathtt{t}\,(\varphi\,[\,x\backslash a\,]\,[\,y\backslash t\,]) \to \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\ [\,y\backslash \mathtt{f}(t)\,].\end{aligned}$$

Now we can get the translation of the existential intro,

$$\begin{aligned}\Gamma.\,\Delta\,[\,x\backslash a\,] \vdash\ &\mathtt{E}(t)\{\ \exists\, y:\sigma\ \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\,,\\ &\quad \mathtt{c}(t) \vdash \exists\,\mathtt{I}(\\ &\qquad y:\sigma \vdash \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\,,\\ &\qquad \mathtt{f}(t),\\ &\qquad \mathtt{s}(\varphi\,[\,x\backslash a\,]; y\backslash t)(\mathtt{k}\,(\rho\,[\,x\backslash a\,]; \varphi\,[\,x\backslash a\,]\,[\,y\backslash t\,]))\\ &\quad)\ \}: \exists\, y:\sigma\ \mathtt{t}\,(\varphi\,[\,x\backslash a\,])\end{aligned}$$

which correctly has the typing of the translation of the type inference of the original term. Therefore we finally have translation soundness.

Equality intro, axiom terms
The same way as the other ones.

Definite description proof-term
The main thing to use is that we actually get the translation of the definite description proof-term over the context $\Gamma.\,x:\alpha.\,\Delta$. Apply inductive hypothesis to get the substituted versions of the arguments.

Proof-variable
Not treated here.

The proof is the same for the most part for proof-term substitution except that we don't need to write out as many substitutions. $\square$

Theorem 2.27. All derivable sequents $\Gamma \vdash \mathcal{J}$ with $\imath$-free contexts Γ are translation sound.

Proof. Proceed by induction on trees of derivation. We begin with handling the structural rules. For weakening,

$$\frac{\Gamma.\,\Delta \vdash \mathcal{J} \qquad \Gamma \vdash \alpha\ \texttt{type}}{\Gamma.\,x : \alpha.\,\Delta \vdash \mathcal{J}}$$

where we immediately get translation soundness because our translation does not depend on the context. For instance, if $\mathcal{J} = \varphi\ \texttt{prop}$ then by inductive hypothesis we have that

$$\Gamma.\,\Delta \vdash \texttt{t}\,(\varphi)\ \ \texttt{prop}$$

$$\Gamma.\,\Delta.\,p : \texttt{t}\,(\varphi) \vdash \texttt{i}(\varphi)(p) : \varphi \qquad\qquad \Gamma.\,\Delta.\,p : \varphi \vdash \texttt{j}(\varphi)(p) : \texttt{t}\,(\varphi)$$

and since $\Gamma \vdash \alpha\ \texttt{type}$ is derivable we can apply weakening and derive the desired sequents,

$$\Gamma.\,x : \alpha.\,\Delta \vdash \texttt{t}\,(\varphi)\ \ \texttt{prop},$$

$$\Gamma.\,x : \alpha.\,\Delta.\,p : \texttt{t}\,(\varphi) \vdash \texttt{i}(\varphi)(p) : \varphi, \qquad\qquad \Gamma.\,x : \alpha.\,\Delta.\,p : \varphi \vdash \texttt{j}(\varphi)(p) : \texttt{t}\,(\varphi)\,.$$

For substitution,

$$\frac{\Gamma.\,x : \alpha.\,\Delta \vdash \mathcal{J} \qquad \Gamma \vdash t : \alpha}{\Gamma.\,\Delta\,[\,x \backslash t\,] \vdash \mathcal{J}\,[\,x \backslash t\,]}$$

we apply Lemma 2.26. The symbol rules are also translation soundness preserving as already observed. For the equality rules coming from symbol rules they follow from the fact that translation soundness gives us strong enough assumptions to directly apply the rule itself. For the following equality rule,

$$\frac{\Gamma \vdash t : \sigma \qquad \Gamma \vdash \tau : \sigma \qquad \Gamma \vdash \rho : t_\sigma \tau}{\Gamma \vdash t = \tau : \sigma}$$

the translation soundness of the premises gives us the translation soundness of the conclusion.

As all the rules of the type theory are translation soundness preserving, then all the $\imath$-free sequents are translation sound. $\square$

This concludes the most labour intensive part of handling the syntax.

2.3.4 Translating sequents

With the translation of judgements in $\imath$-free contexts set up it is an easier task to translate entire sequents. We will define the translation $\mathtt{t}(\Gamma)$ together with a substitution $\mathtt{i}(\Gamma)$ with the intention that

$$\mathtt{t}(\Gamma) \vdash \mathtt{i}(\Gamma) : \Gamma \qquad\qquad \Gamma \vdash \mathtt{j}(\Gamma) : \mathtt{t}(\Gamma)$$

Definition 2.28. Define the translation of a context $\mathtt{t}(\Gamma)$ together with its context substitutions $\mathtt{i}(\Gamma)$ and $\mathtt{j}(\Gamma)$ by induction on the length of Γ and based on case analysis on whether the rightmost type is a proposition or a sort,

$$\begin{array}{ll} \mathtt{t}(\Gamma.\, x : \sigma) = \mathtt{t}(\Gamma).\, x : \sigma & \mathtt{t}(\Gamma.\, p : \varphi) = \mathtt{t}(\Gamma).\, p : \mathtt{t}\,(\varphi\,[\mathtt{i}(\Gamma)]) \\ \mathtt{i}(\Gamma.\, x : \sigma) = \mathtt{i}(\Gamma).x & \mathtt{i}(\Gamma.\, p : \varphi) = \mathtt{i}(\Gamma).\mathtt{i}(\varphi\,[\mathtt{i}(\Gamma)])(p) \\ \mathtt{j}(\Gamma.\, x : \sigma) = \mathtt{j}(\Gamma).x & \mathtt{j}(\Gamma.\, p : \varphi) = \mathtt{j}(\Gamma).\mathtt{j}(\varphi\,[\mathtt{i}(\Gamma)])(p) \end{array}$$

Proposition 2.29. If Γ is a raw context such that $\Gamma \vdash \mathtt{ctxt}$ is derivable then

(i) $\mathtt{t}(\Gamma) \vdash \mathtt{ctxt}$, $\mathtt{t}(\Gamma) \vdash \mathtt{i}(\Gamma) : \Gamma$ and $\mathtt{t}(\Gamma) \vdash \mathtt{j}(\Gamma) : \Gamma$ are derivable,

(ii) $\mathtt{t}(\Gamma)$ is an $\imath$-free context.

Proof. Straightforward induction on the length of Γ, applying Theorem 2.27 when the last type is a proposition. □

Proposition 2.30. The category $\mathcal{I}_{\Sigma,\mathcal{T}}$ is equivalent to the full subcategory $\mathcal{D}_{\Sigma,\mathcal{T}}$ consisting only of $\imath$-free contexts.

Proof. We construct the equivalence by first showing that, as context morphisms, $\mathtt{i}(\Gamma)$ and $\mathtt{j}(\Gamma)$ are inverse to eachother. On the sort components of Γ it is immediately clear as both $\mathtt{i}(\Gamma)$ and $\mathtt{j}(\Gamma)$ just take the literal variable. On the propositional components it follows from the fact that any two proof-terms of the same proposition are judged equal.

Next we see that they are both equal to the identity when Γ is $\imath$-free. It is clear on sort components and on propositional components this follows because when $p : \varphi$ in Γ is $\imath$-free we have that $\mathtt{t}\,(\varphi) = \varphi$ and any two proof-terms of φ are equal.

This exhibits $\mathcal{D}_{\Sigma,\mathcal{T}}$ as a retract equivalent of $\mathcal{I}_{\Sigma,\mathcal{T}}$. □

The next step is translating back and forth between this $\imath$-free fragment of the type theory and the corresponding first order logic. We will take small intermediate steps by structuring up the contexts of the type theory using the following facts about type theory.

Proposition 2.31. The order of parts of a context which don't depend on each other does not matter, i.e., if $\Gamma \vdash \Delta\ \mathtt{ctxt}$, $\Gamma \vdash \Delta'\ \mathtt{ctxt}$ and $\Gamma.\,\Delta.\,\Delta' \vdash \mathcal{J}$ are derivable then so is $\Gamma.\,\Delta'.\,\Delta \vdash \mathcal{J}$.

Proof. By induction on the length of Δ. First when Δ has length 1, assume that $\Gamma \vdash \alpha$ $\texttt{type}$ and $\Gamma \vdash \Delta'$ $\texttt{ctxt}$ (it does not matter whether $\texttt{type}$ is $\texttt{sort}$ or $\texttt{prop}$). Also assume that $\Gamma.\, x : \alpha.\, y : \Delta' \vdash \mathcal{J}$ is derivable. Then by weakening, $\Gamma.\, x : \alpha.\, y : \Delta'.\, z : \alpha \vdash \mathcal{J}$ is derivable, but then by substitution, $\Gamma.\, y : \Delta'.\, z : \alpha \vdash \mathcal{J}\,[\,x \backslash z\,]$. $\square$

Proposition 2.32. If $\Gamma.\, p : \varphi \vdash \psi$ $\texttt{prop}$ or $\Gamma.\, p : \varphi \vdash t : \sigma$ is derivable where φ is a proposition and ψ, t are $\imath$-free then $\Gamma \vdash \psi$ $\texttt{prop}$ or $\Gamma \vdash t : \sigma$ is derivable.

Proof. Induction on the structure of ψ, t. The only sort-terms have a proof-term subexpression is the $\imath$-term and propositions have either other propositions or sort-terms as subexpressions. $\square$

This allows us to translate each sequent into one where the sorts are to the left and the propositions are to the right.

Proposition 2.33. For each derivable $\imath$-free sequent $\Gamma \vdash \mathcal{J}$ we have that

$$\Gamma_{\texttt{sort}}.\, \Gamma_{\texttt{prop}} \vdash \mathcal{J}$$

is derivable, where $\Gamma_{\texttt{sort}}$ is the subcontext of sorts and $\Gamma_{\texttt{prop}}$ is the subcontext of propositions.

Proposition 2.34. The category $\mathcal{D}_{\Sigma,\mathcal{T}}$ is equivalent to the category $\mathcal{E}_{\Sigma,\mathcal{T}}$ which is the full subcategory of $\mathcal{D}_{\Sigma,\mathcal{T}}$ whose objects are those contexts of the form $\Gamma_{\texttt{sort}}.\, \Gamma_{\texttt{prop}}$.

2.4 The syntactic equivalence

In this section we will define an equivalence of categories by defining functors between the syntactic category of the first order theory $\mathbb{C}_{\Sigma,\mathcal{T}}$ and the $\imath$-free term category $\mathcal{E}_{\Sigma,\mathcal{T}}$. This will be made by pairing $\imath$-free sequents $\Gamma_{\texttt{sort}}.\, \Gamma_{\texttt{prop}} \vdash \rho : \varphi$ with first-order sequents, and then giving a proposition corresponding to a context morphism which can be translated to a functional relation.

We first define the pairing by induction on the structure of φ.

Definition 2.35. Define the translation of $\imath$-free propositions into first order logic by letting

$$\begin{array}{lcl}
\texttt{fol}(\top) & = & \top \\
\texttt{fol}(\bot) & = & \bot \\
\texttt{fol}(\varphi \vee \psi) & = & \varphi \vee \psi \\
\texttt{fol}(\exists\, x : \sigma\ \psi) & = & \exists\, x : \sigma\ \psi \\
\texttt{fol}(\forall\, x : \sigma\ \psi) & = & \forall\, x : \sigma\ \psi \\
\texttt{fol}(\exists\, p : \varphi\ \psi) & = & \varphi \wedge \psi \\
\texttt{fol}(\forall\, p : \varphi\ \psi) & = & \varphi \Rightarrow \psi \\
\texttt{fol}(\Xi(\vec{\tau})) & = & \Xi(\vec{\tau})
\end{array}$$

and translate $\Gamma_{\mathtt{prop}}$ by $\mathtt{fol}(\Gamma_{\mathtt{prop}}.\, p : \varphi) = \mathtt{fol}(\Gamma_{\mathtt{prop}}, \mathtt{fol}(\varphi))$. Also call the inverse of the above map $\mathtt{fol}^{-1}$.

Theorem 2.36. A $\imath$-free sequent $\Gamma_{\mathtt{sort}}.\, \Gamma_{\mathtt{prop}} \vdash \rho : \varphi$ is derivable in $\mathscr{FOT}(\Sigma, \mathcal{T})$ iff the sequent $\Gamma_{\mathtt{sort}} \mid \mathtt{fol}(\Gamma_{\mathtt{prop}}) \vdash \mathtt{fol}(\varphi)$ is derivable in first order logic, and an $\imath$-free sequent $\Gamma_{\mathtt{sort}}.\, \Gamma_{\mathtt{prop}} \vdash t : \sigma$ for a sort-term t is derivable iff the sequent $\Gamma_{\mathtt{sort}} \mid t : \sigma$ is derivable in first order logic.

Proof. Proof by induction on trees of derivation. We can basically pair the symbol rules of the type theory with rules from first order logic and check that they are equivalent. We have a perfect match for the $\imath$-free terms. To get equivalence for propositions we use that the propositions from first order logic are admissible in the sense that for example if

$$\Gamma \mid \varphi \wedge \psi \ \mathtt{prop}$$

is derivable then

$$\Gamma \mid \varphi \ \mathtt{prop} \qquad\qquad \Gamma \mid \psi \ \mathtt{prop}$$

are derivable, to get a perfect match for most of the symbol rules. Some care is needed to the rules involving conjunction and implication, but here we utilize Proposition 2.32 to get that the rules are equivalent. ☐

Now we will define

$$F : \mathcal{E}_{\Sigma,\mathcal{T}} \to \mathbb{C}_{\Sigma,\mathcal{T}},$$

first F_0 on objects

$$F_0(\Gamma_{\mathtt{sort}}.\, \Gamma_{\mathtt{prop}}) = \{\Gamma_{\mathtt{sort}} \mid \mathtt{fol}(\Gamma_{\mathtt{prop}})\}.$$

For $F_0(\Gamma)$ to be an object of the syntactic category we need that $\Gamma_{\mathtt{sort}} \mid \varphi \ \mathtt{prop}$ for each component φ of $\mathtt{fol}(\Gamma_{\mathtt{prop}})$ but this is the case by Theorem 2.36. Next we define the action F_1 on a context morphism $f : \Gamma' \to \Gamma$ by using the equality predicate on the context $\Gamma_{\mathtt{sort}}.\, \Gamma_{\mathtt{prop}} = \Gamma$ to

$$[\mathtt{fol}(\mathtt{t}\,(f\,[\mathsf{x}] =_{\Gamma} \mathtt{y}))] : \{\Gamma'_{\mathtt{sort}} \mid \mathtt{fol}(\Gamma'_{\mathtt{prop}})\} \to \{\Gamma_{\mathtt{sort}} \mid \mathtt{fol}(\Gamma_{\mathtt{prop}})\}$$

where $\mathtt{x} : \Gamma'.\, \mathtt{y} : \Gamma \vdash f\,[\mathsf{x}] =_{\Gamma} \mathtt{y} \ \mathtt{prop}$ is derivable in the type theory. It is a functional relation by the properties of the equality predicate and by soundness of translation. Also, it respects the judgemental equality of the type theory, because if $\Gamma \vdash f = g : \Gamma'$ then the propositions $f\,[\mathsf{x}] =_{\Gamma} \mathtt{y}$ and $g\,[\mathsf{x}] =_{\Gamma} \mathtt{y}$ are equivalent. For the other direction G, we let

$$G_0\{\Gamma \mid \Theta\} = \Gamma.\, \mathtt{fol}^{-1}(\Theta)$$

where soundness once again is used to prove that $\Gamma \vdash \theta$ for each component θ of $\mathtt{fol}^{-1}(\Theta)$. On arrows $[\phi] : \{\Gamma \mid \Theta\} \to \{\Delta \mid \Lambda\}$ we take

$$G_1[\phi] = \imath\ (\Delta.\, \mathtt{fol}^{-1}(\Lambda))\ \mathtt{fol}^{-1}(\phi)(\varepsilon, \upsilon)$$

where ε and υ are given by ϕ being a functional relation together with proof-terms witnessing those things in the type theory by Theorem 2.36. It takes equivalent functional relations to judgementally equal context morphisms because if ψ is equivalent to ϕ then

$$\Gamma.\, \vec{x} : \Delta.\, \vec{y} : \Delta \mid \psi\, [\vec{x}] =_{\Delta} \phi\, [\vec{y}]$$

is derivable which means that they are taken to propositionally equal context morphisms in the type theory, but then we get judgemental equality.

3 Semantics

3.1 Modelling type theory

A standard way of modelling type theory is with a so-called Category with Families. Here is a definition.

Definition 3.1. A *category with families* consists of

(i) a category $\mathcal{C}$ with a (specific) terminal object $\circledast$, called its base category,

(ii) two presheaves Type and Tm on $\mathcal{C}$, called the type and term presheaves,

(iii) a natural transformation $\mathsf{tm} : \mathsf{Tm} \to \mathsf{Type}$,

and an operation taking an object Γ in $\mathcal{C}$ and $A \in \mathsf{Type}(\Gamma)$ to the following pullback diagram in the presheaf category over $\mathcal{C}$,

$$\begin{tikzcd} \Gamma\centerdot A \arrow[r, "q_{\Gamma,A}"] \arrow[d, "p_{\Gamma,A}"'] & \mathsf{Tm} \arrow[d, "\mathsf{tm}"] \\ \Gamma \arrow[r, "`A'"'] & \mathsf{Type} \end{tikzcd}$$

where Γ and $\Gamma.A$ are Yoneda embedded objects of $\mathcal{C}$ and 'A' is induced by $A \in \mathsf{Type}(\Gamma)$ with the Yoneda lemma. The data of that diagram is called the comprehension of A in Γ.

We will also be using the following notation.

- The arrow action on presheaves will be denoted from the right, so if $f : \Gamma' \to \Gamma$ and $A \in \mathsf{Type}(\Gamma)$, we will denote $\mathsf{Type}(f)(A) \in \mathsf{Type}(\Gamma')$ by $A\{f\}$. Note that because of contravariance, $A\{g \circ f\} = A\{g\}\{f\}$.
- Instead of writing $\mathsf{tm}^{-1}(\Gamma, A)$ where $A \in \mathsf{Type}(\Gamma)$, we will write $\mathsf{Tm}(\Gamma, A)$.
- Although the data of a judgement fibration actually is given by a triple $(\mathsf{Type}, \mathsf{Tm}, \mathsf{tm})$, we will most of the time only be writing Type. Confusion should not arise because we will not consider different judgement fibrations with the same presheaf for Type.
- The names for judgement fibrations we will be using are Prop (or $(\mathsf{Prop}, \mathsf{Pf}, \mathsf{pf})$) and Sort (or $(\mathsf{Sort}, \mathsf{Tm}, \mathsf{tm})$).
- We will usually not be writing out the context in $p_{\Gamma,A}$, preferring just to write p_A.
- By the universal property of the pullback, arrows '('t) $: \Gamma \to \mathsf{Tm}$ not only correspond to elements $t \in \mathsf{Tm}(\Gamma, A)$ (Yoneda lemma), they also correspond to sections $\ulcorner t \urcorner : \Gamma \to \Gamma\centerdot A$ of p_A.

For an expository treatment, see Hofmann [2]. We will work with several such structures on the same category but the above terminology is unsuitable for talking about that. Keeping with the proud tradition of introducing different names for concepts similar to categories with families, we will use the following terminology.

Definition 3.2. A *judgment fibration* on a category $\mathcal{C}$ with terminal object $\circledast$ is a category with families whose base category is $(\mathcal{C}, \circledast)$.

We call it a fibration because the presheaves on a category are equivalent to the discrete fibrations on that same category. We will not prefer one of these perspectives over the other throughout the text, rather, we will use whichever seems more convenient to the author at that moment.

3.1.1 Logical structure on judgement fibrations

Now we will define the structure needed to model $\mathscr{FOT}(\Sigma, \mathcal{T})$ with judgement fibrations. These semantics are those provided by the framework in [7] although we will present them slightly differently. We will pick out some properties familiar from functorial semantics using the following suggestive notation.

Definition 3.3. A judgement fibration Prop over $\mathcal{C}$ is said to be *proof irrelevant* if given any $\Gamma \in \mathcal{C}$, $\varphi \in \mathsf{Prop}(\Gamma)$ and $\rho, \delta \in \mathsf{Pf}(\Gamma, \varphi)$, ρ and δ are identical elements of $\mathsf{Pf}(\Gamma, \varphi)$.

Remark 3.4. We have the following facts about proof irrelevant judgement fibrations:

- A judgement fibration is proof irrelevant iff the map $\mathsf{pf} : \mathsf{Pf} \to \mathsf{Prop}$ is mono.
- Any pair of arrows 'ρ', 'δ' $: \Gamma \to \mathsf{Pf}$ for $\rho, \delta \in \mathsf{Pf}(\Gamma, \varphi)$ where Prop is a proof irrelevant judgement fibration has Γ as equalizer.

Definition 3.5. A judgement fibration pair $\mathsf{Prop}, \mathsf{Sort}$ over $\mathcal{C}$ is said to carry an *equality structure* if there is a function natural in Γ assigning

(i) to each $\sigma \in \mathsf{Sort}(\Gamma)$, $t, \tau \in \mathsf{Tm}(\Gamma, \sigma)$

$$t =_\sigma \tau \in \mathsf{Prop}(\Gamma)$$

(ii) to each $\sigma \in \mathsf{Sort}(\Gamma)$, $t \in \mathsf{Tm}(\Gamma, \sigma)$

$$=\!\mathsf{I}(\sigma, t) \in \mathsf{Pf}(\Gamma, t =_\sigma t)$$

(iii) such that if $\rho \in \mathsf{Pf}(\Gamma, t =_\sigma \tau)$ then t and τ are identical elements of $\mathsf{Tm}(\Gamma, \sigma)$.

Proposition 3.6. For an equality structure on $\mathsf{Prop}, \mathsf{Sort}$, the arrow

$$p_{(t=_\sigma \tau)} : \Gamma.\,(t =_\sigma \tau) \to \Gamma$$

is a weak equalizer for

$$\ulcorner t \urcorner, \ulcorner \tau \urcorner : \Gamma \to \Gamma.\,\sigma$$

given any $\sigma \in \mathsf{Sort}(\Gamma)$ and $t, \tau \in \mathsf{Tm}(\Gamma, \sigma)$.

Proof. Assume that we have an equality structure. Clauses (ii) and (iii) of Definition 3.5 together with a little diagram chasing gives us a section s of

$$p_{t=_\sigma \tau} : \Gamma.\,(t =_\sigma \tau) \to \Gamma$$

precisely when $\ulcorner t \urcorner = \ulcorner \tau \urcorner$. This is the equalizing condition for the diagram

$$\begin{tikzcd}
& \Gamma \arrow[dl, "s"'] \arrow[d, "\mathbf{1}"] \arrow[dr, shift left] \arrow[dr, shift right] & \\
\Gamma.\,(t =_\sigma \tau) \arrow[r, "p_\sigma"] & \Gamma \arrow[r, shift left, "\ulcorner t \urcorner"] \arrow[r, shift right, "\ulcorner \tau \urcorner"'] & \Gamma.\,\sigma
\end{tikzcd}$$

which we want to generalize to any $f : \Gamma' \to \Gamma$ in $\mathcal{C}$. Given such an arrow, we can use naturality by pulling back along f and get a section of

$$p_{(t=_\sigma \tau)\{f\}} : \Gamma'.\,(t =_\sigma \tau)\{f\} \to \Gamma'$$

precisely when $\ulcorner t \urcorner \circ f = \ulcorner \tau \urcorner \circ f$. But as

$$\begin{tikzcd}
\Gamma'.\,(t =_\sigma \tau)\{f\} \arrow[r] \arrow[d] & \Gamma.\,t =_\sigma \tau \arrow[d] \\
\Gamma' \arrow[r, "f"] & \Gamma
\end{tikzcd}$$

is pullback by the universal property of the pullback such sections correspond uniquely to maps $f' : \Gamma' \to \Gamma.\,(t =_\sigma \tau)$ where $f = p_{t=_\sigma \tau} \circ f'$. $\square$

Definition 3.7. Given a judgement fibration $\mathsf{pf} : \mathsf{Pf} \to \mathsf{Prop}$ over $\mathcal{C}$ and given any $\varphi, \psi \in \mathsf{Prop}(\Gamma)$, we denote the space of morphisms

$$\Gamma.\,\varphi \to \Gamma.\,\psi \text{ over } \Gamma \quad \text{by} \quad \varphi \leq_\Gamma \psi.$$

Remark 3.8. Even though $\varphi \leq_\Gamma \psi$ may contain many elements in general, when Prop is proof irrelevant it contains at most one element and behaves like a proposition. As this is the case we will take interest in, we use this notation.

Definition 3.9. A judgement fibration Prop over $\mathcal{C}$ can carry a *truthity structure*, which consists of functions natural in $\Gamma \in \mathcal{C}$ assigning

(i) an element $\top \in \mathsf{Prop}(\Gamma)$,

(ii) an element $\top\mathsf{I} \in \mathsf{Pf}(\Gamma, \top)$.

Definition 3.10. A judgement fibration pair $\mathsf{Prop}, \mathsf{Sort}$ over $\mathcal{C}$ can carry a *left-adjoint structure*, which consists of functions natural in $\Gamma \in \mathcal{C}$ assigning

(i) to each $\varphi \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \varphi)$

$$\exists_{\mathsf{P}}(\varphi, \psi) \in \mathsf{Prop}(\Gamma),$$

(ii) to each $\varphi \in \mathsf{Prop}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \varphi)$, $\rho \in \mathsf{Pf}(\Gamma, \varphi)$ and $\delta \in \mathsf{Pf}(\Gamma, \psi\{\ulcorner \rho \urcorner\})$

$$\exists_{\mathsf{P}}\mathsf{I}(\varphi, \psi, \rho, \delta) \in \mathsf{Pf}(\Gamma, \exists_{\mathsf{P}}(\varphi, \psi)),$$

(iii) to each $\varphi, \vartheta \in \mathsf{Prop}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \varphi)$, $\rho \in \mathsf{Pf}(\Gamma, \exists_{\mathsf{P}}(\varphi, \psi))$ and $\delta \in \mathsf{Pf}(\Gamma . \varphi . \psi, \vartheta\{p_\varphi\}\{p_\psi\})$

$$\exists_{\mathsf{P}}\mathsf{E}(\varphi, \psi, \vartheta, \rho, \delta) \in \mathsf{Pf}(\Gamma, \vartheta),$$

(iv) to each $\sigma \in \mathsf{Sort}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \sigma)$

$$\exists_{\mathsf{S}}(\sigma, \psi) \in \mathsf{Prop}(\Gamma),$$

(v) to each $\sigma \in \mathsf{Sort}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \sigma)$, $t \in \mathsf{Tm}(\Gamma, \sigma)$ and $\delta \in \mathsf{Pf}(\Gamma, \psi\{\ulcorner t \urcorner\})$

$$\exists_{\mathsf{S}}\mathsf{I}(\varphi, \psi, t, \delta) \in \mathsf{Pf}(\Gamma, \exists_{\mathsf{S}}(\varphi, \psi)),$$

(vi) and to each $\sigma \in \mathsf{Sort}(\Gamma)$, $\vartheta \in \mathsf{Prop}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \sigma)$, $\rho \in \mathsf{Pf}(\Gamma, \exists_{\mathsf{S}}(\varphi, \psi))$ and $\delta \in \mathsf{Pf}(\Gamma . \sigma . \psi, \vartheta\{p_\sigma\}\{p_\psi\})$

$$\exists_{\mathsf{S}}\mathsf{E}(\sigma, \psi, \vartheta, \rho, \delta) \in \mathsf{Pf}(\Gamma, \vartheta).$$

Proposition 3.11. For a judgement fibration pair $\mathsf{Prop}, \mathsf{Sort}$ over $\mathcal{C}$ with a left-adjoint structure, we have for each $\Gamma \in \mathcal{C}$ and $\varphi, \vartheta \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \varphi)$ functions

(i) $\psi \leq_{\Gamma . \varphi} \vartheta\{p_\varphi\} \to \exists_{\mathsf{P}}(\varphi, \psi) \leq_\Gamma \vartheta$,

(ii) $\exists_{\mathsf{P}}(\varphi, \psi) \leq_\Gamma \vartheta \to \psi \leq_{\Gamma . \varphi} \vartheta\{p_\varphi\}$,

and for each $\sigma \in \mathsf{Sort}(\Gamma)$, $\vartheta \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \sigma)$ functions

(iii) $\psi \leq_{\Gamma . \sigma} \vartheta\{p_\sigma\} \to \exists_{\mathsf{S}}(\sigma, \psi) \leq_\Gamma \vartheta$,

(iv) $\exists_{\mathsf{S}}(\sigma, \psi) \leq_\Gamma \vartheta \to \psi \leq_{\Gamma . \sigma} \vartheta\{p_\sigma\}$.

Proving this is a matter of diagram chasing with pullbacks. We will not provide the proof to this proposition, or the similar propositions that follow, here.

Definition 3.12. A judgement fibration Prop can carry a *falsity structure*, which consists of functions natural in $\Gamma \in \mathcal{C}$ assigning

(i) an element $\bot \in \mathsf{Prop}(\Gamma)$,

(ii) to each $\rho \in \mathsf{Pf}(\Gamma, \bot)$ and $\varphi \in \mathsf{Prop}(\Gamma)$,

$$\bot\mathsf{E}(\varphi, \rho) \in \mathsf{Pf}(\Gamma, \varphi).$$

Definition 3.13. A judgement fibration Prop can carry a *disjunction structure*, which consists of functions natural in $\Gamma \in \mathcal{C}$ assigning

(i) to each $\varphi, \psi \in \mathsf{Prop}(\Gamma)$,

$$\vee(\varphi, \psi) \in \mathsf{Prop}(\Gamma),$$

(ii) to each $\varphi, \psi \in \mathsf{Prop}(\Gamma)$ and $\rho \in \mathsf{Pf}(\Gamma, \varphi)$

$$\vee\mathsf{I}_L(\varphi, \psi, \rho) \in \mathsf{Pf}(\vee(\varphi, \psi), \Gamma),$$

(iii) to each $\varphi, \psi \in \mathsf{Prop}(\Gamma)$ and $\rho \in \mathsf{Pf}(\Gamma, \psi)$

$$\vee\mathsf{I}_R(\varphi, \psi, \rho) \in \mathsf{Pf}(\vee(\varphi, \psi), \Gamma),$$

(iv) and to each $\varphi, \psi, \vartheta \in \mathsf{Prop}(\Gamma)$, $\rho \in \mathsf{Pf}(\Gamma\,\text{.}\,\varphi, \vartheta\{p_\varphi\})$, $\delta \in \mathsf{Pf}(\Gamma\,\text{.}\,\psi, \vartheta\{p_\psi\})$ and $\pi \in \mathsf{Pf}(\Gamma, \vartheta)$

$$\vee\mathsf{E}(\varphi, \psi, \vartheta, \rho, \delta, \pi) \in \mathsf{Pf}(\Gamma, \vee(\varphi, \psi)).$$

Proposition 3.14. For a judgement fibration Prop over $\mathcal{C}$ with a disjunction structure, we have for each $\Gamma \in \mathcal{C}$ and $\varphi, \psi, \vartheta \in \mathsf{Prop}(\Gamma)$ functions

(i) $\vartheta \leq_\Gamma \varphi \to \vartheta \leq_\Gamma \vee(\varphi, \psi)$ and $\vartheta \leq_\Gamma \psi \to \vartheta \leq_\Gamma \vee(\varphi, \psi)$,

(ii) $(\varphi \leq_\Gamma \vartheta) \times (\psi \leq_\Gamma \vartheta) \to \vee(\varphi, \psi) \leq_\Gamma \vartheta$.

Definition 3.15. A judgement fibration pair $\mathsf{Prop}, \mathsf{Sort}$ over $\mathcal{C}$ can carry a *right-adjoint structure*, which consists of functions natural in $\Gamma \in \mathcal{C}$ assigning

(i) to each $\varphi \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma\,\text{.}\,\varphi)$

$$\forall_{\mathsf{P}}(\varphi, \psi) \in \mathsf{Prop}(\Gamma),$$

(ii) to each $\varphi \in \mathsf{Prop}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma\,\text{.}\,\varphi)$ and $\rho \in \mathsf{Pf}(\Gamma\,\text{.}\,\varphi, \psi)$

$$\forall_{\mathsf{P}}\mathsf{I}(\varphi, \psi, \rho) \in \mathsf{Pf}(\Gamma, \forall_{\mathsf{P}}(\varphi, \psi)),$$

(iii) and to each $\varphi \in \mathsf{Prop}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma\,\text{.}\,\varphi)$, $\rho \in \mathsf{Pf}(\Gamma, \varphi)$ and $\delta \in \mathsf{Pf}(\Gamma, \psi\{\ulcorner\rho\urcorner\})$

$$\forall_{\mathsf{P}}\mathsf{E}(\varphi, \psi, \rho, \delta) \in \mathsf{Pf}(\Gamma, \psi),$$

(iv) to each $\sigma \in \mathsf{Sort}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \sigma)$

$$\forall_{\mathsf{S}}(\sigma, \psi) \in \mathsf{Prop}(\Gamma),$$

(v) to each $\sigma \in \mathsf{Sort}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \sigma)$ and $\rho \in \mathsf{Pf}(\Gamma . \sigma, \psi)$

$$\forall_{\mathsf{S}}\mathsf{I}(\sigma, \psi, \rho) \in \mathsf{Pf}(\Gamma, \forall_{\mathsf{S}}(\sigma, \psi)),$$

(vi) and to each $\sigma \in \mathsf{Sort}(\Gamma)$, $\psi \in \mathsf{Prop}(\Gamma . \sigma)$, $t \in \mathsf{Tm}(\Gamma, \sigma)$ and $\delta \in \mathsf{Pf}(\Gamma, \psi\{\ulcorner t \urcorner\})$

$$\forall_{\mathsf{S}}\mathsf{E}(\sigma, \psi, t, \delta) \in \mathsf{Pf}(\Gamma, \psi).$$

Proposition 3.16. For a judgement fibration pair $\mathsf{Prop}, \mathsf{Sort}$ over $\mathcal{C}$ with a right-adjoint structure, we have for each $\Gamma \in \mathcal{C}$ and $\varphi, \vartheta \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \varphi)$ functions

(i) $\vartheta\{p_\varphi\} \leq_{\Gamma . \varphi} \psi \to \vartheta \leq_\Gamma \forall_{\mathsf{P}}(\varphi, \psi)$,

(ii) $\vartheta \leq_\Gamma \forall_{\mathsf{P}}(\varphi, \psi) \to \vartheta\{p_\varphi\} \leq_{\Gamma . \varphi} \psi$,

and for each $\sigma \in \mathsf{Sort}(\Gamma)$, $\vartheta \in \mathsf{Prop}(\Gamma)$ and $\psi \in \mathsf{Prop}(\Gamma . \sigma)$ functions

(iii) $\vartheta\{p_\sigma\} \leq_{\Gamma . \sigma} \psi \to \vartheta \leq_\Gamma \forall_{\mathsf{S}}(\sigma, \psi)$,

(iv) $\vartheta \leq_\Gamma \forall_{\mathsf{S}}(\sigma, \psi) \to \vartheta\{p_\sigma\} \leq_{\Gamma . \sigma} \psi$.

Definition 3.17. A unique-choice structure on Prop, Sort consists of functions natural in $\Gamma \in \mathcal{C}$ assigning to each $\sigma \in \mathsf{Sort}(\Gamma)$, $\varphi \in \mathsf{Prop}(\Gamma . \sigma)$, $\varepsilon \in \mathsf{Pf}(\Gamma, \exists_{\mathsf{S}}(\sigma, \varphi))$ and $\upsilon \in \mathsf{Pf}(\Gamma . \sigma . \varphi . \sigma\{p_\sigma\}\{p_\varphi\} . \varphi\{p_\varphi\}\{p_{\sigma\{p_\sigma\}\{p_\varphi\}}\})$

$$\imath\,(\sigma, \varphi, \varepsilon, \upsilon) \in \mathsf{Tm}(\Gamma, \sigma) \quad \text{and} \quad \partial\,(\sigma, \varphi, \varepsilon, \upsilon) \in \mathsf{Pf}(\Gamma, \varphi\{\ulcorner \imath\,(\sigma, \varphi, \varepsilon, \upsilon) \urcorner\}).$$

Remark 3.18. The pairs of elements $x \in \mathsf{Tm}(\Gamma, \sigma)$ and $\rho \in \mathsf{Pf}(\Gamma, \varphi\{x\})$ are equivalent to sections of the map

$$p_\varphi \circ p_\sigma : \Gamma . \sigma . \varphi \to \Gamma.$$

Definition 3.19. We bundle up the above defined structures by the term *logical structure*, i.e., a logical structure is one of the following,

- an equality structure,
- a truthity structure,
- a left-adjoint structure,
- a disjunction structure,
- a falsity structure,
- a right-adjoint structure,

- or a unique-choice structure.

And thusly we have defined the structures in play. Let us now turn to the morphisms between these structures. We will be short here but see Newstead [6] for a more detailed account of these morphisms.

Definition 3.20. A morphism from a judgement pair $\mathsf{Prop}_{\mathcal{C}}$, $\mathsf{Sort}_{\mathcal{C}}$ over $\mathcal{C}$ to a pair $\mathsf{Prop}_{\mathcal{D}}$, $\mathsf{Sort}_{\mathcal{D}}$ over $\mathcal{D}$ consists of a

(i) *base morphism* basepoint-preserving functor $F : \mathcal{C} \to \mathcal{D}$

(ii) natural maps above $\mathcal{C}$

$$F_{\mathsf{Prop}} : \mathsf{Prop}_{\mathcal{C}} \to \mathsf{Prop}_{\mathcal{D}} \circ F \qquad F_{\mathsf{Pf}} : \mathsf{Pf}_{\mathcal{C}} \to \mathsf{Pf}_{\mathcal{D}} \circ F$$
$$F_{\mathsf{Sort}} : \mathsf{Sort}_{\mathcal{C}} \to \mathsf{Sort}_{\mathcal{D}} \circ F \qquad F_{\mathsf{Tm}} : \mathsf{Tm}_{\mathcal{C}} \to \mathsf{Tm}_{\mathcal{D}} \circ F$$

(iii) such that the following diagrams commute,

$$\begin{tikzcd} \mathsf{Pf}_{\mathcal{C}} \arrow[r, "F_{\mathsf{Pf}}"] \arrow[d, "\mathsf{pf}_{\mathcal{C}}"] & \mathsf{Pf}_{\mathcal{D}} \circ F \arrow[d, "\mathsf{pf}_{\mathcal{D}} \circ F"] \\ \mathsf{Prop}_{\mathcal{C}} \arrow[r, "F_{\mathsf{Prop}}"] & \mathsf{Prop}_{\mathcal{D}} \circ F \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Tm}_{\mathcal{C}} \arrow[r, "F_{\mathsf{Tm}}"] \arrow[d, "\mathsf{tm}_{\mathcal{C}}"] & \mathsf{Tm}_{\mathcal{D}} \circ F \arrow[d, "\mathsf{tm}_{\mathcal{D}} \circ F"] \\ \mathsf{Sort}_{\mathcal{C}} \arrow[r, "F_{\mathsf{Prop}}"] & \mathsf{Sort}_{\mathcal{D}} \circ F \end{tikzcd}$$

(iv) and such that for any $\Gamma \in \mathcal{C}$, $\varphi \in \mathsf{Prop}_{\mathcal{C}}(\Gamma)$ and $\sigma \in \mathsf{Sort}_{\mathcal{C}}(\Gamma)$ the induced maps

$$F(\Gamma . \varphi) \to F(\Gamma) . F_{\mathsf{Prop}}(\varphi) \qquad F(\Gamma . \sigma) \to F(\Gamma) . F_{\mathsf{Sort}}(\sigma)$$

are isomorphisms.

We will sometimes hide the subscripts in F_{Prop} and F_{Sort}.

Remark 3.21. The notation used here suggests that all the four comprehensions above are the same but this is not the case. Each comprehension is made in a different judgement fibration.

Definition 3.22. A morphism of judgement fibration pairs is said to preserve a logical structure if both the domain and codomain pairs are equipped with the corresponding structure and if the morphism is equivariant with respect to these structure maps. This means for instance that the morphism F preserves disjunction structures if the following equations are satisfied for any input data,

$$F \vee^{\mathcal{C}} (\varphi, \psi) = \vee^{\mathcal{D}}(F\varphi, F\psi) \qquad F \vee \mathsf{I}^{\mathcal{C}}_L(\varphi, \psi, \rho) = \vee\mathsf{I}^{\mathcal{D}}_L(F\varphi, F\psi, F\rho)$$

$$F \vee \mathsf{I}^{\mathcal{C}}_L(\varphi, \psi, \delta) = \vee\mathsf{I}^{\mathcal{D}}_L(F\varphi, F\psi, F\delta)$$

$$F \vee \mathsf{E}^{\mathcal{C}}(\varphi, \psi, \vartheta, \pi, \rho, \delta) = \vee\mathsf{E}^{\mathcal{D}}(F\varphi, F\psi, F\vartheta, F\pi, F\rho, F\delta)$$

Definition 3.23. A 2-morphism between morphisms F and G of judgement fibration pairs is a natural transformation η between their respective base morphisms such that, for each $\Gamma \in \mathcal{C}$, $\varphi \in \mathsf{Prop}_{\mathcal{C}}(\Gamma)$ and $\sigma \in \mathsf{Sort}_{\mathcal{C}}(\Gamma)$

$$F(\varphi) = G(\varphi)\{\eta_\Gamma\}, \qquad F(\sigma) = G(\sigma)\{\eta_\Gamma\}.$$

3.1.2 Signature and axiom structures

We will now define the relevant structures for interpreting the signature and axioms of a first order theory.

Definition 3.24. A *signature structure* on $\mathcal{C}$ of Σ are functions natural in $\Gamma \in \mathcal{C}$

(i) for each sort σ of Σ assigning an element

$$[\![\sigma]\!] \in \mathsf{Sort}(\Gamma)$$

(ii) for each function symbol f with $\texttt{arity}\, f = (\sigma_1, \ldots, \sigma_n, \sigma)$ assigning to terms $\tau_i \in \mathsf{Tm}(\Gamma, \sigma_i)$ an element

$$[\![f]\!](\vec{\tau}) \in \mathsf{Tm}(\Gamma, [\![\sigma]\!])$$

(iii) for each atomic formula Ξ with $\texttt{arity}\, \Xi = (\sigma_1, \ldots, \sigma_n)$ assigning to terms $\tau_i \in \mathsf{Tm}(\Gamma, \sigma_i)$ an element

$$[\![\Xi]\!](\vec{\tau}) \in \mathsf{Prop}(\Gamma)$$

This is sufficient structure to interpret each derivable sequent

$$\vdash \Gamma\ \texttt{ctxt} \qquad \Gamma \vdash \varphi\ \texttt{prop} \qquad \Gamma \vdash \sigma\ \texttt{sort}$$

$$\Gamma \vdash t : \sigma \qquad \Gamma \vdash \rho : \varphi$$

of the signature Σ as elements

$$[\![\Gamma]\!] \in \mathcal{C} \qquad [\![\varphi]\!] \in \mathsf{Prop}([\![\Gamma]\!]) \qquad [\![\sigma]\!] \in \mathsf{Sort}([\![\Gamma]\!])$$

$$[\![t]\!] \in \mathsf{Tm}([\![\Gamma]\!], [\![\sigma]\!]) \qquad [\![\rho]\!] \in \mathsf{Pf}([\![\Gamma]\!], [\![\varphi]\!])$$

by interpreting the logical symbols with logical structures and interpreting the signature symbols with signature structures. Now for interpreting the axioms,

Definition 3.25. An *axiom structure* on $\mathcal{C}$ of Σ are for each axiom $\mid \Theta \vdash \vartheta$ an assignment of $\rho_i \in \mathsf{Pf}(\Gamma, [\![\theta_i]\!])$ an element

$$[\![\mathcal{A}]\!](\vec{\rho}) \in \mathsf{Pf}(\Gamma, [\![\vartheta]\!])$$

Proposition 3.26. Here are some facts about signature and axiom structures of Σ, $\mathcal{T}$ to make our life easier down the road.

(i) Each function symbol is completely described by an arrow in $\mathcal{C}$, in the sense that if we define $\Gamma.\, [\![\vec{\sigma}]\!]_\Gamma \in \mathcal{C}$ for every $\Gamma \in \mathcal{C}$ by repeatedly extending Γ by the interpretations of σ_i, we get a commutative diagram

$$\begin{tikzcd}
\Gamma. \llbracket \vec{\sigma} \rrbracket_\Gamma \arrow[rr] \arrow[drrr] \arrow[ddr] & & \Gamma. \llbracket \sigma \rrbracket_\Gamma \arrow[drrr] \arrow[ddl] & & & \\
& & & \circledast. \llbracket \vec{\sigma} \rrbracket_\circledast \arrow[rr, "C_f"] \arrow[ddr] & & \circledast. \llbracket \sigma \rrbracket_\circledast \arrow[ddl] \\
& \Gamma \arrow[drrr] & & & & \\
& & & & \circledast &
\end{tikzcd}$$

where all the square sides are pullback and the subscript on the interpretations denote which context the interpretation was evaluated in. This yields a natural bijection between the parametrization of $\llbracket f \rrbracket(\vec{\tau}) \in \mathsf{Tm}(\Gamma, \llbracket \sigma \rrbracket)$ by $\tau_i \in \mathsf{Tm}(\Gamma, \llbracket \sigma \rrbracket_i)$ and the parametrization of $C_f \circ g$ by arrows $g : \Gamma \to \circledast. \llbracket \vec{\sigma} \rrbracket_\circledast$.

(ii) An axiom structure can similarly be classified by an arrow (necessarily mono if Prop is proof irrelevant)

$$\circledast. \llbracket \Theta \rrbracket_\circledast \to \circledast. \llbracket \vartheta \rrbracket_\circledast.$$

We finally arrive at the definition of model.

Definition 3.27. A model $\mathcal{M}$ of $\mathscr{FOT}(\Sigma, \mathcal{T})$ is a judgement fibration pair with logical structures, a signature structure and an axiom structure. The morphisms of models are the morphisms of judgement fibration pairs that are equivariant with respect to these structures. The 2-morphisms of models are the 2-morphisms of judgement fibration pairs.

3.1.3 A comment on the framework

To see how this presentation of the semantics connects to those given by Uemura in [7], let us take a specific look at the disjunction structure. By the magic of discrete fibrations, the functions specifying a disjunction structure perfectly correspond to morphisms of discrete fibrations over $\mathcal{C}$. For instance, the function $\vee(\varphi, \psi)$ corresponds to morphism $\mathsf{Prop} \times \mathsf{Prop} \to \mathsf{Prop}$ over $\mathcal{C}$. Further, the left and right intro functions correspond to discrete fibrations $\mathcal{L}_{\vee \mathtt{I}_L}, \mathcal{L}_{\vee \mathtt{I}_R}$ which are limits over the diagrams

$$
\begin{tikzcd}
\mathcal{L}_{\vee\mathtt{I}_L} & & \\
\mathsf{Prop} & & \\
\mathsf{Pf} \arrow[d] & \mathsf{Prop}^2 \arrow[ul, "\vee"'] \arrow[dl] \arrow[r] & \mathsf{Prop} \\
\mathsf{Prop} & &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \mathcal{L}_{\vee\mathtt{I}_R} & \\
\mathsf{Prop} & & \mathsf{Pf} \arrow[d] \\
 & \mathsf{Prop}^2 \arrow[ul, "\vee"'] \arrow[dl] \arrow[r] & \mathsf{Prop} \\
\mathsf{Prop} & &
\end{tikzcd}
$$

where the maps $\mathsf{Pf} \to \mathsf{Prop}$ are pf. We also have morphisms $\vee\mathsf{I}_L, \vee\mathsf{I}_R$ over the diagram making the following commute

$$
\begin{tikzcd}
 & \mathsf{Pf} \arrow[dd, dashed] & \mathcal{L}_{\vee\mathtt{I}_R} \arrow[l, bend right, "\vee\mathsf{I}_R"'] & \\
\mathcal{L}_{\vee\mathtt{I}_L} \arrow[ur, bend left, "\vee\mathsf{I}_L"] & & & \\
 & \mathsf{Prop} & & \mathsf{Pf} \arrow[d] \\
 & \mathsf{Pf} \arrow[d] & \mathsf{Prop}^2 \arrow[ul, "\vee"'] \arrow[dl] \arrow[r] & \mathsf{Prop} \\
 & \mathsf{Prop} & &
\end{tikzcd}
$$

and the elimination rule corresponds similarly to a discrete fibration $\mathcal{L}_{\vee\mathtt{E}}$ which is a limit of the following diagram except for the dashed morphism. It comes with a morphism $\vee\mathsf{E}$ over $\mathcal{C}$ the diagram commute,

$$\begin{tikzcd}
\mathcal{L}_{\vee\mathtt{E}} \arrow[rrrdd, bend left, "\vee\mathsf{E}"] & & & & \\
& \mathsf{Pf} \arrow[d] & & & \mathsf{abs}_{\mathsf{pf}}\mathsf{Pf} \arrow[d] \\
& \mathsf{Prop} & & \mathsf{Pf} \arrow[ddd, dashed] & \mathsf{abs}_{\mathsf{pf}}\mathsf{Prop} \arrow[dl] \\
\mathsf{abs}_{\mathsf{pf}}\mathsf{Pf} \arrow[d] & & \mathsf{Prop}^2 \arrow[ul, "\vee"'] \arrow[r] \arrow[dl] & \mathsf{Prop} & \\
\mathsf{abs}_{\mathsf{pf}}\mathsf{Prop} \arrow[r] & \mathsf{Prop} & & & \mathsf{Prop}^2 \arrow[ul] \arrow[uu] \arrow[dl] \\
& & \mathsf{Prop}^2 \arrow[ul] \arrow[ull] \arrow[r] & \mathsf{Prop} &
\end{tikzcd}$$

here the $\mathsf{abs}_{\mathsf{pf}}$ is the polynomial functor given by pf which models abstraction over a proposition, the arrow into $\mathsf{abs}_{\mathsf{pf}}\mathsf{Prop}$ models weakening and the arrow out of $\mathsf{abs}_{\mathsf{pf}}\mathsf{Prop}$ picks out which proposition it is abstracting over.

So $\mathcal{L}_{\vee\mathtt{I}_L}, \mathcal{L}_{\vee\mathtt{I}_R}, \mathcal{L}_{\vee\mathtt{E}}$ model type theoretic contexts from a semantic type theory $\mathbb{T}$ given in [7] and the morphisms $\vee\mathsf{I}_L, \vee\mathsf{I}_R, \vee\mathsf{E}$ induce morphisms from $\mathcal{L}_{\vee\mathtt{I}_L}, \mathcal{L}_{\vee\mathtt{I}_R}, \mathcal{L}_{\vee\mathtt{E}}$ into the limit of larger diagrams such that they are sections from the projection maps from the limit of the larger diagram into the limit of the smaller diagram.

3.1.4 Logical structures on the standard pair

Here we will introduce the two natural judgement fibrations associated with a Heyting category. First the subobject fibration,

Definition 3.28. Let $\bullet$ denote the final fibration over $\mathcal{C}$ which pointwise picks out a single element and Sub the subobject fibration which over Γ picks out all monos $T \rightarrowtail \Gamma$ modulo isomorphism. This description of Sub only makes it a fibration if Sub has pullbacks of all monos. Let max denote the map $\bullet \to \mathsf{Sub}$ which pointwise picks out the maximal subobject of Γ.

Proposition 3.29. Let $\mathcal{C}$ be a category with pullbacks of all monos. A choice of representative for each subobject $J \subseteq \Gamma$ makes the map $\mathsf{max} : \bullet \to \mathsf{Sub}$ a judgement fibration.

Proof. By the Yoneda lemma, let 'J' $: \Gamma \to \mathsf{Sub}$ pick out the subobject $J \subseteq \Gamma$. Let J be represented by $j : T \rightarrowtail \Gamma$. Then the following diagram is pullback,

$$\begin{tikzcd}
T \arrow[r] \arrow[d, tail, "j"'] & \bullet \arrow[d, "\mathsf{max}"] \\
\Gamma \arrow[r, "`J\text{'}"] & \mathsf{Sub}
\end{tikzcd}$$

which can be proven using the description of subobjects of representable fibrations as sieves. This technical description is beyond the scope of this work, see MacLane and Moerdijk [5, I.4] for a treatment. $\square$

Definition 3.30. Let Ob denote the fibration over $\mathcal{C}$ which pointwise is constantly equal to the set of all objects of $\mathcal{C}$, and Ar which pointwise over Γ picks out the set morphisms out of Γ.

Proposition 3.31. Let $\mathcal{C}$ be a category with binary products. Then $\mathsf{cod} : \mathsf{Ar} \to \mathsf{Ob}$ which pointwise picks out the codomain of each morphism is a judgement fibration.

Proof. By Yoneda lemma, let 'A' $: \Gamma \to \mathsf{Ob}$ pick out the object A in $\mathcal{C}$ and let 'π_2' $: \Gamma \times A \to \mathsf{Ar}$ pick out the morphism $\pi_2 : \Gamma \times A \to A$. Then the following diagram is pullback,

$$\begin{tikzcd}
\Gamma \times A \arrow[r, "`\pi_2\text{'}"] \arrow[d, "\pi_1"'] & \mathsf{Ar} \arrow[d, "\mathsf{cod}"] \\
\Gamma \arrow[r, "`A\text{'}"] & \mathsf{Ob}
\end{tikzcd}$$

which can be seen by considering it pointwise over Γ',

$$\begin{tikzcd}
T \arrow[drr, bend left] \arrow[dd, bend right] & & \\
& \mathrm{Hom}(\Gamma', \Gamma \times A) \arrow[r, "f \mapsto f^*(\pi_2) = \pi_2 \circ f"] \arrow[d, "f \mapsto \pi_1 \circ f"'] & \sum_{X \in \mathrm{Ob}\,\mathcal{C}} \mathrm{Hom}(\Gamma', X) \arrow[d, "\mathsf{cod}"] \\
& \mathrm{Hom}(\Gamma', \Gamma) \arrow[r, "f \mapsto f^*(A) = A"] & \mathrm{Ob}\,\mathcal{C}
\end{tikzcd}$$

where the maps out of T commuting means that the upper map factors through $\mathrm{Hom}(\Gamma', A)$ and by the universal property of the product, we get a unique arrow $T \to \mathrm{Hom}(\Gamma', \Gamma \times A)$ making the diagram commute. $\square$

Definition 3.32. We say that $\mathcal{C}$ has subobject intersections over Γ if for each $\varphi, \psi \subseteq \Gamma$ there is a $\varphi \cap \psi$ with the universal properties that

(i) if $\vartheta \subseteq \Gamma$ satisfies that $\vartheta \leq_\Gamma \varphi$ and $\vartheta \leq_\Gamma \psi$ then $\vartheta \leq_\Gamma \varphi \cap \psi$,

(ii) both $\varphi \cap \psi \leq_\Gamma \varphi$ and $\varphi \cap \psi \leq_\Gamma \psi$.

We say that $\mathcal{C}$ has subobject intersections if it has subobject intersections over all $\Gamma \in \mathcal{C}$

Definition 3.33. We will call $\mathcal{C}$ a *suitable base category* if it has binary products, pullbacks of all monos and a choice of representative for each subobject. We will call the judgement fibration pair Sub, Ob the *standard pair* on $\mathcal{C}$ when $\mathcal{C}$ is a suitable base category.

Proposition 3.34. If $\mathcal{C}$ is a suitable base category then it has subobject intersections and Sub has a truthity structure, and is proof irrelevant.

Proof. If $\varphi, \psi \subseteq \Gamma$ are subobjects of Γ then their intersection is given by the pullback

$$\begin{tikzcd} \Gamma_\bullet \varphi\{p_\psi\} \arrow[r, tail] \arrow[d, tail] & \Gamma_\bullet \varphi \arrow[d, tail, "p_\varphi"] \\ \Gamma_\bullet \psi \arrow[r, tail, "p_\psi"] & \Gamma \end{tikzcd} .$$

and the truthity structure is given by taking $\Gamma \subseteq \Gamma$ for each Γ. It is proof irrelevant because $\bullet$ is a singleton above each Γ. $\square$

Proposition 3.35. The standard pair carries an equality structure iff the base category has equalizers.

Proof. Assume that they carry an equality structure. Take two parallel arrows $f, g : \Gamma' \to \Gamma$. We get two sections $[f, \mathbf{1}], [g, \mathbf{1}]$ of the context projection $\Gamma' \times \Gamma \to \Gamma$ which correspond to two elements $t, \tau \mathsf{Tm}(\Gamma', \Gamma)$. By Proposition 3.6 we get $\ulcorner t \urcorner, \ulcorner \tau \urcorner$ have a weak equalizer if $\mathsf{Sub}, \mathsf{Sort}$ carry an equality structure. By proof irrelevance and the weak equalizer being the extension of Γ' by $t =_\Gamma \tau$ (see Remark 3.4), the equalizing arrow $\Gamma'_\bullet t =_\Gamma \tau \to \Gamma'$ is a mono. Therefore the weak equalizer is in fact an equalizer.

Now assume that the base category has equalizers. Assign to two parallel arrows the subobject given by their equalizer. We get the structure maps of an equality structure and naturality by the fact that any two equalizers of the same map induce equal subobjects. $\square$

Definition 3.36. If $\mathcal{C}$ is a suitable base category we say that it has subobject intersections if for all $\Gamma \in \mathcal{C}$, $\varphi \subseteq \Gamma$ and $\psi \subseteq \Gamma$, there is a $\varphi \cup \psi \subseteq \Gamma$ such that

$$\varphi \leq_\Gamma \varphi \cup \psi, \qquad \psi \leq_\Gamma \varphi \cup \psi, \qquad \text{if } \varphi \leq_\Gamma \vartheta \text{ and } \psi \leq_\Gamma \vartheta \text{ then } \varphi \cup \psi \leq_\Gamma \vartheta$$

and if $f : \Gamma' \to \Gamma$ then $(\varphi \cup \psi)\{f\} = \varphi\{f\} \cup \psi\{f\}$.

Definition 3.37. If $\mathcal{C}$ is a suitable base category we say that it has initial subobjects if for every Γ there is a $\bot_\Gamma \subseteq \Gamma$ such that for every $\vartheta \subseteq \Gamma$ we have $\bot_\Gamma \leq_\Gamma \vartheta$, and for any $f : \Gamma' \to \Gamma$ we have that $\bot_{\Gamma'} = \bot_\Gamma\{f\}$.

Lemma 3.38. We characterize disjunction and falsity structures for the standard pair.

- Disjunction structures carried by the standard pair are equivalent to subobject unions in $\mathcal{C}$.

- Falsity structures carried by the standard pair are equivalent to initial subobjects in $\mathcal{C}$.

Proof. First note that the universal properties of subobject unions and initial subobjects define them uniquely when they exist so they are mere properties of $\mathcal{C}$. Therefore we do not need to check that the constructions that will constitute our equivalence are mutually inverse.

Proposition 3.14 together with the naturality condition in Definition 3.13 means that a disjunction structure gives us subobject unions. To see that subobject unions give us a disjunction structure, note that the universal property of the subobject unions gives us the following for each $\Gamma \in \mathcal{C}$,

(i) for each $\varphi, \psi \subseteq \Gamma$, a subobject $\varphi \cup \psi \subseteq \Gamma$,

(ii) such that if φ is maximal in Γ then so is $\varphi \cup \psi$,

(iii) and if ψ is maximal in Γ then so is $\varphi \cup \psi$,

(iv) and whenever a subobject $\vartheta \subseteq \Gamma$ has that both

$$\vartheta\{p_\varphi\} \subseteq \Gamma.\,\varphi \text{ and } \vartheta\{p_\psi\} \subseteq \Gamma.\,\psi$$

are maximal and $\varphi \cup \psi \subseteq \Gamma$ is maximal, then so is ϑ.

The final clause follows because $\vartheta\{p_\varphi\}$ is maximal precisely when $\varphi \leq_\Gamma \vartheta$. Naturality of this assignment needs only be checked for (i), it then follows by proof irrelevance for (ii)-(iv). But it holds immediately for (i) by definition.

The procedure is similar for a falsity structure. $\square$

Proposition 3.39. Every morphism in a suitable base category factors as a mono followed by a product projection.

Proof. A morphism $f : \Gamma' \to \Gamma$ together with $\mathbf{1} : \Gamma' \to \Gamma'$ induces an arrow $[f, \mathbf{1}] : \Gamma' \to \Gamma \times \Gamma'$ that is a mono. $\square$

Lemma 3.40. Now we exploit the above factorization.

- Left-adjoint structures carried by the standard pair are equivalent to dependent sums for subobjects in $\mathcal{C}$.

- Right-adjoint structures carried by the standard pair are equivalent to dependent products for subobjects in $\mathcal{C}$.

Proof. Both items work much the same way so let us treat the first one. First let us note that given $\varphi \in \mathsf{Sub}(\Gamma)$, $\psi \in \mathsf{Sub}(\Gamma')$ and $f : \Gamma' \to \Gamma$ the condition on $\vartheta \in \mathsf{Sub}(\Gamma)$ that

$$\psi \leq_{\Gamma'} \varphi\{f\} \quad \Leftrightarrow \quad \vartheta \leq_\Gamma \varphi$$

uniquely determines the subobject ϑ if it exists. Therefore, by Proposition 3.11, if the standard pair carries a left-adjoint structure and $\mathcal{C}$ has dependent subobject sums then they coincide.

Also note that if $\mathcal{C}$ has dependent subobject sums then we get an induced left-adjoint structure: We can take the dependent subobject sum along monos and product projections to define

(i) for subobjects $i : \varphi \rightarrowtail \Gamma$ and $j : \psi \rightarrowtail \varphi$ let $\exists_{\mathsf{P}}(\varphi, \psi) = \Sigma_i(\psi)$

(ii) which means that if φ is maximal in Γ and $\psi\{i^{-1}\}$ is maximal in Γ then so is $\Sigma_i(\psi)$ in Γ

(iii) and if $\Sigma_i(\psi)$ is maximal in Γ then for any $k : \vartheta \rightarrowtail \Gamma$ such that $\vartheta\{j \circ i\}$ is maximal in ψ then so is ϑ in Γ

where the two latter clauses hold by the universal properties of dependent sums, and

(iv) for each objects Γ, σ of $\mathcal{C}$ and $j : \psi \rightarrowtail \Gamma \times \sigma$ let $\exists_{\mathsf{S}}(\sigma, \psi) = \Sigma_{\pi_1}(\psi)$

(v) which means that for any map $f : \Gamma \to \sigma$ that makes $\psi\{[\mathbf{1}, f]\}$ maximal in Γ then $\Sigma_{\pi_1}(\psi)$ is also maximal in Γ

(vi) and if $\Sigma_{\pi_1}(\psi)$ is maximal in Γ then for any $k : \vartheta \rightarrowtail \Gamma$ such that $\vartheta\{j \circ \pi_1\}$ is maximal in ψ then so is ϑ in Γ

and the same is true for the final two clauses. This precisely corresponds to the clauses of Definition 3.10 with $\mathsf{Prop} = \mathsf{Sub}$ and $\mathsf{Pf} = \bullet$. Let us end by expanding on how the universal properties of the dependent sums give us the last two clauses.

For (v) note that the unit of the adjunction gives us $\psi \leq_{\Gamma \times \Sigma} \Sigma_{\pi_1}(\psi)\{\pi_1\}$ which we can pull back both along $[\mathbf{1}, f]$ to get

$$\psi\{[\mathbf{1}, f]\} \leq_\Gamma \Sigma_{\pi_1}(\psi)\{\pi_1 \circ [\mathbf{1}, f]\}$$

but as $[\mathbf{1}, f]$ is a section of π_1 and $\psi\{[\mathbf{1}, f]\}$ is maximal in Γ the clause is satisfied.

For (vi) note that $\vartheta\{j \circ \pi_1\}$ is maximal in ψ precisely when

$$\psi \leq_\Gamma \vartheta\{\pi_1\}$$

which by adjointness means that

$$\Sigma_{\pi_1}(\psi) \leq_\Gamma \vartheta$$

and since Σ_{π_1} is maximal in Γ then so is ϑ. □

Lemma 3.41. If the standard pair carries a left-adjoint and an equality structure then it also carries a (necessarily unique) unique-choice structure.

Proof. If it has a left-adjoint structure and an equality structure then the base category is regular and the equality predicate is modelled by equalizers. With a left-adjoint structure then whenever $\exists_{\mathsf{S}}(\sigma, \varphi) \subseteq \Gamma$ is maximal then the map

$$u = p_\sigma \circ p_\varphi : \Gamma \centerdot \sigma \centerdot \varphi \to \Gamma \centerdot \sigma \to \Gamma$$

is a regular epimorphism. And with an equality structure then whenever $(v_1 =_\sigma v_2) \subseteq \Gamma \centerdot \sigma \centerdot \varphi \centerdot \sigma \centerdot \varphi$ is maximal then u is a monomorphim. But as maps that are both regular epi and mono are isomorphisms, we can take its (necessarily unique) inverse to get a section of $u : \Gamma \centerdot \sigma \centerdot \varphi$. This gives us the structure maps of a unique-choice structure, and the unicity of the inverse ensures that the structure maps are natural so we indeed have a unique-choice structure. $\square$

3.2 Recovering functorial semantics for first order logic

We will now show how any Heyting functor out of $\mathcal{I}_{\Sigma,\mathcal{T}}$ corresponds to a first-order morphism out of $\mathcal{I}_{\Sigma,\mathcal{T}}$.

Lemma 3.42. A judgement fibration pair morphism from $\mathcal{I}_{\Sigma,\mathcal{T}}$ to a standard pair is equivariant with respect to unique-choice structures.

Proof. Any functor preserves isomorphisms. $\square$

Theorem 3.43. Each Heyting functor $F : \mathcal{I}_{\Sigma,\mathcal{T}} \to \mathcal{C}$ induces uniquely a first-order morphism $(F, F_{\mathsf{Prop}}, F_{\mathsf{Sort}}) : \mathcal{I}_{\Sigma,\mathcal{T}} \to \mathcal{C}$ for a standard pair on $\mathcal{C}$.

Proof. As a Heyting functor is a functor that preserves finite limits and is equivariant with respect to finite subobject unions, dependent sums and dependent products. In particular the image of $\lozenge$, the selected terminal object of $\mathcal{I}_{\Sigma,\mathcal{T}}$, is taken to a terminal object $\circledast$ which we take as our selected terminal object of the base category.

Let us describe the judgement fibration pair morphism. For the propositional part, we get F_{Prop} by taking the proposition over Γ given by $\Gamma \vdash \varphi$ `prop` to the subobject given by $F(\Gamma.\, p : \varphi) \rightarrowtail F(\Gamma)$ where the monomorphism is preserved because F preserves finite limits, and F_{Pf} is given by noting that whenever $\Gamma \vdash \rho : \varphi$ then Γ is isomorphic to $\Gamma.\, p : \varphi$, so $F(\Gamma.\, p : \varphi)$ yields the maximal subobject of Γ. This gives us by definition that $F(\Gamma.\, p : \varphi) \cong F(\Gamma) \centerdot F(\varphi)$.

Now note that if F preserves finite limits then it preserves terminal objects (truthity structure) and equalizers (equality structure). As it is equivariant with respect to finite subobject unions, dependent sums, and dependent products, we get that it preserves falsity, disjunction, left-adjoint and right-adjoint structures.

Finally, it is equivariant with respect to unique-choice structure.

For the preservation of the signature and axiom structures, the preservation of the signature structure is given by functoriality and the axiom structure is given by functoriality together with cartesian functors preserving monos. $\square$

Theorem 3.44. The base functor of a first-order morphism $(F, F_{\mathsf{Prop}}, F_{\mathsf{Sort}}) : \mathcal{I}_{\Sigma,\mathcal{T}} \to \mathcal{C}$ is a Heyting functor.

Proof. For equalizer preservation, we need to prove it by induction on the length of quantifying context. For product preservation, we need to prove that if $\vdash \Gamma$ `ctxt` and $\vdash \Gamma'$ `ctxt` then $F(\Gamma.\, \Gamma')$ is the product of $F(\Gamma)$ and $F(\Gamma')$.

For finite union preservation we can simply use falsity and disjunction equivariances.

For preservation of dependent sums and dependent products we once again need to do induction on the length of the quantifying context. $\square$

This takes us to the main result of this work, showing that we recover the functors of the functorial semantics within the semantic framework of Uemura. Let us first see the precise statement that we are relying upon.

Theorem 3.45 (Uemura [7, Theorem 6.9].)**.** Let $\mathcal{M}$ be a model of $\mathscr{FOT}(\Sigma, \mathcal{T})$. The category of morphisms of models $\mathscr{FOT}(\Sigma, \mathcal{T}) \to \mathcal{M}$ is contractible.

Contractibility of this functor category means that there is at least one such functor, and that any other functor must be isomorphic to that one. But as we will see now, if two morphisms $\mathscr{FOT}(\Sigma, \mathcal{T}) \to \mathcal{M}$ are isomorphic then they are equal.

Lemma 3.46. Let $\mathcal{M}$ be a standard model of $\mathscr{FOT}(\Sigma, \mathcal{T})$ on a suitable base category $\mathcal{C}$ and let F, G be two morphisms of models $\mathcal{I}_{\Sigma,\mathcal{T}} \to \mathcal{M}$. If F and G are naturally isomorphic in the 2-categorical sense of Definition 3.23 then they are equal.

Proof. Assume $\eta_\Gamma : F \Rightarrow G$ is such an isomorphism. We need to prove that

(i) $F(\Gamma) = G(\Gamma)$,

(ii) $F_{\texttt{sort}}(\Gamma, \sigma) = G_{\texttt{sort}}(\Gamma, \sigma)$,

(iii) $F_{\texttt{prop}}(\Gamma, \varphi) = G_{\texttt{prop}}(\Gamma, \varphi)$.

We will need to proceed by induction on the length of Γ. Let us just first note that because whenever $\Gamma \vdash \sigma$ `sort` is derivable we have $\Diamond \vdash \sigma$ `sort` we get

$$F_{\texttt{sort}}(\Gamma, \sigma) = F_{\texttt{sort}}(\Diamond, \sigma)\{!_{F(\Gamma)}\}$$

where $!_{F(\Gamma)}$ denotes the morphism to the terminal object. But because F is a morphism of models we must have $F_{\texttt{sort}}(\Diamond, \sigma) = [\![\sigma]\!]_{\circledast}$ and similar for G.

Let us now proceed with the induction. Because the base functors preserve basepoint we have $F(\Diamond) = G(\Diamond) = \circledast$.

Now let Γ be such that $F(\Gamma) = G(\Gamma)$. We get that

(i) $F_{\texttt{sort}}(\Gamma, \sigma) = [\![\sigma]\!]_{\circledast}\{!_{F(\Gamma)}\} = [\![\sigma]\!]_{\circledast}\{!_{G(\Gamma)}\} = G_{\texttt{sort}}(\Gamma, \sigma)$

(ii) $F_{\texttt{prop}}(\Gamma, \varphi) \subseteq F(\Gamma)$ is isomorphic to $G_{\texttt{prop}}(\Gamma, \varphi) \subseteq G(\Gamma)$ over $F(\Gamma) = G(\Gamma)$ because of the action of the natural isomorphism η_Γ on propositions satisfying $G_{\texttt{prop}}(\Gamma, \varphi)\{\eta_\Gamma\} = F_{\texttt{prop}}(\Gamma, \varphi)$. But that means that they are equal as subobjects.

This means that when we extend $F(\Gamma) = G(\Gamma)$ by the image of a proposition or sort, they agree for F and G and therefore the extensions are also equal. $\square$

Theorem 3.47. If $\mathcal{C}$ is a suitable base category, the standard models of

$$\mathscr{FOT}(\Sigma, \mathcal{T})$$

on a category $\mathcal{C}$ correspond bijectively to functorial models of the first order theory Σ, $\mathcal{T}$ in $\mathcal{C}$. $\square$

4 Further directions

Let us conclude with some suggestions on how to develop this line of thought.

With functorial semantics we have a richer universal property for the syntactic category because we not only get that Heyting functors correspond to first-order models, but we also that a certain notion of morphism of models corresponds to natural transformations of Heyting functors.

These do not seem to be captured by the semantics of Uemura [7] although there is a notion of *theory* over a type theory given there which might be developed such that morphisms of models of a theory over a type theory recover the morphisms of models in the functorial sense.

4.1 Semantics for a proof relevant version with homotopy type theory

In this section we will (very broadly) sketch an interpretation of a proof relevant version of $\mathscr{FOT}$ in homotopy type theory. Let us remove the rule for proof irrelevance from $\mathscr{FOT}$ and replace the equality rule

$$\frac{\Gamma \vdash \sigma \ \texttt{sort} \qquad \Gamma \vdash t : \sigma \qquad \Gamma \vdash \tau : \sigma \qquad \Gamma \vdash \rho : t =_\sigma \tau}{\Gamma \vdash t = \tau : \sigma}$$

by a proof-relevant version,

$$\frac{\begin{array}{c}\Gamma \vdash \sigma \ \texttt{sort} \qquad \Gamma . x : \sigma \vdash \varphi \ \texttt{prop} \\ \Gamma \vdash t : \sigma \qquad \Gamma \vdash \tau : \sigma \qquad \Gamma \vdash \alpha : t =_\sigma \tau \qquad \Gamma \vdash \rho : \varphi[x \backslash t]\end{array}}{\Gamma \vdash {=}\mathtt{E}(\sigma, x.\varphi, t, \tau, \alpha, \rho) : \varphi[x \backslash \tau]}$$

As $\mathscr{HTT}$ satisfies weakening and has substitution and none of the terms in the free first order type theory satisfy any equations we can provide a sound model of it by interpreting the symbols of $\mathscr{FOT}$ in a well-typed way in $\mathscr{HTT}$.

Definition 4.1. A *translation model in* $\mathscr{HTT}$ of $\mathscr{FOT}(\Sigma, \mathcal{T})$ is a function $\mathcal{T}$ taking each type judgement form of $\mathscr{FOT}$ to a type judgement form in $\mathscr{HTT}$, each type symbol of $\mathscr{FOT}(\Sigma)$ to a type in $\mathscr{HTT}$ and each term symbol of $\mathscr{FOT}(\Sigma, \mathcal{T})$ to a term in $\mathscr{HTT}$ such that if $\Gamma \vdash \mathcal{J}$ is a derivable judgement of $\mathscr{FOT}(\kappa)$, then $\mathcal{T}(\Gamma) \vdash \mathcal{T}(\mathcal{J})$ is a derivable judgement of $\mathscr{HTT}$ (where an equality judgement in $\mathscr{FOT}$ is taken to an equality judgement in $\mathscr{HTT}$).

The particular translation models we will highlight are those where each sort will be interpreted as an $n+1$-type and each proposition as an n-type.

Definition 4.2. Define the n-*model* of $\mathscr{FOT}$ as the model in $\mathscr{HTT}$ of $\mathscr{FOT}$ given by interpreting the propositional judgement as belonging to the n-types, the sort judgement as belonging to the $n+1$-types, the propositional symbols in the following way:

$\mathcal{T}(\top)$	$\mathbf{1}$
$\mathcal{T}(\bot)$	$\lvert\mathbf{0}\rvert_n$
$\mathcal{T}(\phi \vee \psi)$	$\lvert\mathcal{T}(\phi) + \mathcal{T}(\psi)\rvert_n$
$\mathcal{T}(\forall\, p : \psi\ \varphi)$	$\Pi\, p : \mathcal{T}(\psi)\ \mathcal{T}(\varphi)$
$\mathcal{T}(\exists\, p : \psi\ \varphi)$	$\Sigma\, p : \mathcal{T}(\psi)\ \mathcal{T}(\varphi)$
$\mathcal{T}(\forall\, x : \sigma\ \varphi)$	$\Pi\, x : \mathcal{T}(\sigma)\ \mathcal{T}(\varphi)$
$\mathcal{T}(\exists\, x : \sigma\ \varphi)$	$\lvert\Sigma\, x : \mathcal{T}(\sigma)\ \mathcal{T}(\varphi)\rvert_n$
$\mathcal{T}(x =_\sigma y)$	$x =_{\mathcal{T}(\sigma)} y$

The atomic propositions can straightforwardly be interpreted as an n-type living above the interpretation of its arity. Proof terms can be interpreted as the intro rules and eliminators for the types on the right hand side. Providing a term for the definite descriptions amounts to constructing a term of

$$(\Pi\, x\, x' : A,\ p : \varphi(x)\, p' : \varphi(x')\ x = x) \times (\lvert\Sigma\, x : A\ \varphi\rvert_n) \to \Sigma\, x : A\ \varphi$$

for any n-type φ and $n+1$-type A. We claim that the target actually is an n-type when an element of the first argument is given, which means that we simply can eliminate out of the second term.

Theorem 4.3. Given an n-type $\varphi(x)$ dependent on the type $x : A$ such that we have a term u of type

$$u : \Pi\, x\, x' : A,\ p : \varphi(x)\, p' : \varphi(x')\ x = x$$

we have that for each $x, x' : A$ and $p : \varphi(x)\, p' : \varphi(x')$, $x = x'$ is contractible.

Proof. We proceed by induction on n. We begin by verifying that it holds for the minimal case, $n = -2$. Then $\varphi(x)$ is contractible for every x and we equivalently to the hypothesis we have a term u of type

$$u : \Pi\, x\, x' : A\ x = x'$$

(in other words, we don't need to depend and the terms of φ anymore) from which we can conclude that $x = x'$ is contractible according to Lemma 3.11.10 of the homotopy type theory book [8].

For the induction step we will argue by passing to the loop space and ticking down the homotopy level, allowing us to use the induction hypothesis. First we do some path algebra to set us up. Allow some $x : A$ and $p : \varphi(x)$ together with u to be given. We wish to prove that for any x' and $\alpha : x = x'$ we have an equality

$$\alpha = u^{-1}(x, x, p, p) \cdot u(x, x', p, \mathfrak{t}_{\alpha}\ (p)).$$

We provide it by path induction on α based at x. The expression reduces to

$$\mathrm{refl}_x = u^{-1}(x, x, p, p) \cdot u(x, x, p, \mathfrak{t}_{\mathrm{refl}_x}\ (p))$$

which we certainly can prove. Now, given any $x, x' : A$ and $p : \varphi(x)$, $p' : \varphi(x')$, consider the type

$$\Sigma\, \alpha : x = x' \;\; \mathfrak{t}_{\alpha}\ (p) = p'.$$

The homotopy level of $\mathfrak{t}_{\alpha}\ (p) = p'$ is lower than that of $\varphi(x)$ and given any $\alpha, \alpha' : x = x'$ and $t : \mathfrak{t}_{\alpha}\ (p) = p'$, $t' : \mathfrak{t}_{\alpha'}\ (p) = p'$, we get an equality

$$\begin{aligned}\alpha &= u^{-1}(x, x, p, p) \cdot u(x, x', p, \mathfrak{t}_{\alpha}\ (p)) = \\ &= u^{-1}(x, x, p, p) \cdot u(x, x', p, p') = \\ &= u^{-1}(x, x, p, p) \cdot u(x, x', p, \mathfrak{t}_{\alpha'}\ (p)) = \alpha'\end{aligned}$$

which means that we can apply the induction hypothesis to get that $\alpha = \alpha'$ is contractible. This means that $x = x'$ is a mere proposition. The inhabitant $u(x, x', p, p') : x = x'$ then gives that $x = x'$ is contractible. □

References

[1] Michael P Fourman. The logic of topoi. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 1053–1090. Elsevier, 1977.

[2] Martin Hofmann. Syntax and semantics of dependent types. In *Extensional Constructs in Intensional Type Theory*, pages 13–54. Springer, 1997.

[3] Bart Jacobs. *Categorical logic and type theory*, volume 141. Elsevier, 1999.

[4] Peter T Johnstone. *Sketches of an elephant: A topos theory compendium*, volume 2. Oxford University Press, 2002.

[5] Saunders MacLane and Ieke Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory.* Springer Science & Business Media, 2012.

[6] Clive Newstead. *Algebraic models of dependent type theory.* PhD thesis, Carnegie Mellon University, 2018.

[7] Taichi Uemura. A general framework for the semantics of type theory. *arXiv preprint arXiv:1904.04097*, 2019.

[8] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.