THE JOURNAL OF SYMBOLIC LOGIC
Volume 69, Number 4, Dec. 2004

HIGHER-ORDER SEMANTICS AND EXTENSIONALITY

CHRISTOPH BENZMÜLLER, CHAD E. BROWN, AND MICHAEL KOHLHASE

Abstract. In this paper we re-examine the semantics of classical higher-order logic with the purpose of clarifying the role of extensionality. To reach this goal, we distinguish nine classes of higher-order models with respect to various combinations of Boolean extensionality and three forms of functional extensionality. Furthermore, we develop a methodology of abstract consistency methods (by providing the necessary model existence theorems) needed to analyze completeness of (machine-oriented) higher-order calculi with respect to these model classes.

§1. Motivation. In classical first-order predicate logic, it is rather simple to assess the deductive power of a calculus: first-order logic has a well-established and intuitive set-theoretic semantics, relative to which completeness can easily be verified using, for instance, the abstract consistency method (cf. the introductory textbooks [6, 22]). This well understood meta-theory has supported the development of calculi adapted to special applications—such as automated theorem proving (cf. [16, 47] for an overview).

In higher-order logics, the situation is rather different: the intuitive set-theoretic standard semantics cannot give a sensible notion of completeness, since it does not admit complete (recursively axiomatizable) calculi [24, 6]. There is a more general notion of semantics [26], the so-called Henkin models, that allows complete (recursively axiomatizable) calculi and therefore sets the standard for deductive power of calculi.

Peter Andrews' *Unifying Principle for Type Theory* [1] provides a method of higher-order abstract consistency that has become the standard tool for completeness proofs in higher-order logic, even though it can only be used to show completeness relative to a certain Hilbert style calculus $\mathfrak{T}_\beta$. A calculus $\mathscr{C}$ is called complete relative to a calculus $\mathfrak{T}_\beta$ iff (if and only if) $\mathscr{C}$ proves all theorems of $\mathfrak{T}_\beta$. Since $\mathfrak{T}_\beta$ is not complete with respect to Henkin models, the notion of completeness that can be established by this method is a strictly weaker notion than Henkin completeness. The differences between these notions of completeness can largely be analyzed in terms of availability of various extensionality principles, which can be expressed axiomatically in higher-order logic.

As a consequence of the limitations of Andrew's *Unifying Principle*, calculi for higher-order automated theorem proving [1, 32, 33, 34, 42, 36, 37] and the corresponding theorem proving systems such as TPS [7, 8], or earlier versions of the LEO [14] system are not complete with respect to Henkin models. Moreover, they

Received February 23, 1998; final version March 29, 2004.

0022-4812/04/6904-0004/$7.20

are not even sound with respect to $\mathfrak{T}_\beta$, since they (for the most part) employ η-conversion, which is not admissible in $\mathfrak{T}_\beta$. In other words, their deductive power lies somewhere between $\mathfrak{T}_\beta$ and Henkin models. Characterizing exactly where reveals important theoretical properties of these calculi that have direct consequences for the adequacy in various application domains (see the discussion in section 8.1). Unlike calculi without computational concerns, calculi for mechanized reasoning systems cannot be made complete by simply adding extensionality axioms, since the search spaces induced by their introduction grow prohibitively. Being able to compare and characterize the methods and computational devices used instead is a prerequisite for further development in this area.

In this situation, the aim of this article is to provide a semantical meta theory that will support the development of higher-order calculi for automated theorem proving just as the corresponding methodology does in first-order logic. To reach this goal, we need to establish:

(1) classes of models that adequately characterize the deductive power of existing theorem-proving calculi (providing semantics with respect to which they are sound and complete), and
(2) a methodology of abstract consistency methods (by providing for these model classes the necessary model existence theorems, which extend Andrews' Unifying Principle), so that the completeness analysis for higher-order calculi will become almost as simple as in first-order logic.

We fully achieve the first goal in this article, and take a large step towards the second. In the model existence theorems presented in this article, we have to assume a new condition called *saturation*, which limits their utility in completeness proofs for machine-oriented calculi. Fortunately, the saturation condition can be lifted by extensions of the methods presented in this article (see the discussion in the conclusion 8.2 and [12]).

Due to the inherent complexity of higher-order semantics we first give an informal exposition of the issues covered and the techniques applied. In Section 4, we will investigate the properties of the model classes introduced in Section 3 in more detail and corroborate them with example models in Section 5. We prove model existence theorems for the model classes in Section 6. Finally, in Section 7 we will apply the model existence theorems from Section 6 to the task of proving completeness of higher-order natural deduction calculi. Section 8 concludes the article with a discussion of related work, possible applications, and the saturation assumption we introduced for the model existence theorems.

The work reported in this article is based on [15] and significantly extends the material presented there.

§2. Informal exposition.

Before we turn to the exposition of the semantics in Section 2.3, let us specify what we mean by "higher-order logic": any simply typed logical system that allows quantification over function and predicate variables. Technically, we will follow tradition and employ a logical system $\mathcal{HOL}$ based on the simply typed λ-calculus as introduced in [18]; this does not restrict the generality of the methods reported in this article, since the ideas can be carried over. A related logical system is discussed in detail in [6].

2.1. Simply typed λ-calculus. To formulate higher-order logic we start with a collection of types $\mathscr{T}$. We assume there are some basic types in $\mathscr{T}$ and that whenever $\alpha, \beta \in \mathscr{T}$, then the function type $(\alpha \to \beta)$ is in $\mathscr{T}$. Furthermore, we assume the types are generated freely, so that $(\alpha_1 \to \beta_1) \equiv (\alpha_2 \to \beta_2)$ implies $\alpha_1 \equiv \alpha_2$ and $\beta_1 \equiv \beta_2$.

$\mathscr{HOL}$-formulae (or *terms*) are built up from a set $\mathscr{V}$ of (typed) variables and a *signature* Σ (a set of typed constants) as *applications* and λ-*abstractions*. We assume the set $\mathscr{V}_\alpha$ of variables of type α is countably infinite for each type α. The set $\mathrm{wff}_\alpha(\Sigma)$ of *well-formed formulae* consists of those formulae which have type α. The type of formula $\boldsymbol{A}_\alpha$ will be annotated as an index, if it is not clear from the context. We will denote variables with upper-case letters $(X_\alpha, Y, Z, X^1_\beta, X^2_\gamma, \dots)$, constants with lower-case letters $(c_\alpha, f_{\alpha\to\beta}, \dots)$ and well-formed formulae with upper-case bold letters $(\boldsymbol{A}_\alpha, \boldsymbol{B}, \boldsymbol{C}^1, \dots)$. Finally, we abbreviate multiple applications and abstractions in a kind of vector notation, so that $\boldsymbol{A}\overline{\boldsymbol{U}^k}$ denotes k-fold application (associating to the left), $\lambda\overline{X^k}\boldsymbol{.}\boldsymbol{A}$ denotes k-fold λ-abstraction (associating to the right) and we use the square dot '$\boldsymbol{.}$' as an abbreviation for a pair of brackets, where '$\boldsymbol{.}$' stands for the left one with its partner as far to the right as is consistent with the bracketing already present in the formula. We may avoid full bracketing of formulas in the remainder if the bracketing structure is clear from the context.

We will use the terms like *free* and *bound* variables or *closed* formulae in their standard meaning and use $\mathrm{free}(\boldsymbol{A})$ for the set of free variables of a formula $\boldsymbol{A}$. In particular, alphabetic change of names of bound variables is built into $\mathscr{HOL}$: we consider alphabetic variants to be identical (viewing the actual representation as a representative of an alphabetic equivalence class) and use a notion of substitution that avoids variable capture by systematically renaming bound variables.[1] We denote a substitution that instantiates a free variable X with a formula $\boldsymbol{A}$ with $[\boldsymbol{A}/X]$ and write $\sigma, [\boldsymbol{A}/X]$ for the substitution that is identical with σ but instantiates X with $\boldsymbol{A}$. For any term $\boldsymbol{A}$ we denote by $\boldsymbol{A}[\boldsymbol{B}]_p$ the term resulting by replacing the subterm at position p in $\boldsymbol{A}$ by $\boldsymbol{B}$.

A structural equality relation of $\mathscr{HOL}$ terms is induced by $\beta\eta$-*reduction*

$$(\lambda X\boldsymbol{.}\boldsymbol{A})\boldsymbol{B} \to_\beta [\boldsymbol{B}/X]\boldsymbol{A} \qquad\qquad (\lambda X\boldsymbol{.}\boldsymbol{C}X) \to_\eta \boldsymbol{C}$$

where X is not free in $\boldsymbol{C}$. It is well-known that the reduction relations β, η, and $\beta\eta$ are terminating and confluent on $\mathrm{wff}(\Sigma)$, so that there are unique normal forms (cf. [9] for an introduction). We will denote the β-normal form of a term $\boldsymbol{A}$ by $\boldsymbol{A}{\downarrow_\beta}$, and the $\beta\eta$-normal form of $\boldsymbol{A}$ by $\boldsymbol{A}{\downarrow_{\beta\eta}}$. If we allow both reduction and expansion steps, we obtain notions of β-*conversion*, η-*conversion*, and $\beta\eta$-*conversion*. We say $\boldsymbol{A}$ and $\boldsymbol{B}$ are β-*equal* [η-*equal*, $\beta\eta$-*equal*] (written $\boldsymbol{A}{\equiv_\beta}\boldsymbol{B}$ [$\boldsymbol{A}{\equiv_\eta}\boldsymbol{B}$, $\boldsymbol{A}{\equiv_{\beta\eta}}\boldsymbol{B}$]) when $\boldsymbol{A}$ is β-convertible [η-convertible, $\beta\eta$-convertible] to $\boldsymbol{B}$.

2.2. Higher-order logic ($\mathscr{HOL}$). In $\mathscr{HOL}$, the set of base types is $\{o, \iota\}$ for truth values and individuals. We will call a formula of type o a *proposition*, and a *sentence* if it is closed. We will assume that the signature Σ contains logical constants for *negation* $(\neg_{o\to o})$, *disjunction* $(\vee_{o\to o\to o})$, and *universal quantification* $(\Pi^\alpha_{(\alpha\to o)\to o})$ for each type α. Optionally, Σ may contain *primitive equality* $(=^\alpha_{\alpha\to\alpha\to o})$ for each type

[1]We could also have used de Bruijn's indices [19] as a concrete implementation of this approach at the syntax level.

α. All other constants are called *parameter*s, since the argumentation in this article is parametric in their choice.

We write disjunctions and equations, i.e., terms of the form $((\vee\boldsymbol{A})\boldsymbol{B})$ or $((=\boldsymbol{A})\boldsymbol{B})$, in infix notation as $\boldsymbol{A} \vee \boldsymbol{B}$ and $\boldsymbol{A} = \boldsymbol{B}$. As we only assume the logical constants $\neg$, $\vee$, and Π^α (and possibly $=^\alpha$) as primitive, we will use formulae of the form $\boldsymbol{A} \wedge \boldsymbol{B}$, $\boldsymbol{A} \Rightarrow \boldsymbol{B}$, and $\boldsymbol{A} \Leftrightarrow \boldsymbol{B}$ as shorthand for the formulae $\neg((\neg\boldsymbol{A}) \vee (\neg\boldsymbol{B}))$, and $(\neg\boldsymbol{A}) \vee \boldsymbol{B}$, and $(\boldsymbol{A} \Rightarrow \boldsymbol{B}) \wedge (\boldsymbol{B} \Rightarrow \boldsymbol{A})$, respectively. For each $\boldsymbol{A} \in \mathrm{wff}_o(\Sigma)$, the standard notations $\forall X_\alpha . \boldsymbol{A}$ and $\exists X_\alpha . \boldsymbol{A}$ for quantification are regarded as shorthand for $\Pi^\alpha(\lambda X_\alpha . \boldsymbol{A})$ and $\neg(\Pi^\alpha(\lambda X_\alpha . \neg\boldsymbol{A}))$. Finally, we extend the vector notation for λ-binders to k-fold quantification: we will use $\forall\overline{X^k}.\boldsymbol{A}$ and $\exists\overline{X^k}.\boldsymbol{A}$ in the obvious way.

We often need to distinguish between atomic and non-atomic formulae in $\mathrm{wff}_o(\Sigma)$. A non-atomic formula is any formula whose β-normal form is either of the form $\neg\boldsymbol{A}$, $\boldsymbol{A} \vee \boldsymbol{B}$, or $\Pi^\alpha\boldsymbol{C}$ (where $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_o(\Sigma)$ and $\boldsymbol{C} \in \mathrm{wff}_{\alpha\to o}(\Sigma)$). An atomic formula is any other formula in $\mathrm{wff}_o(\Sigma)$—including primitive equations $\boldsymbol{A} =^\alpha \boldsymbol{B}$ in case of the presence of primitive equality.

It is matter of folklore that equality can directly be expressed in $\mathscr{HOL}$. A prominent example is the *Leibniz formula* for equality

$$\mathbf{Q}^\alpha := (\lambda X_\alpha Y_\alpha . \forall P_{\alpha\to o} . PX \Rightarrow PY).$$

With this definition, the formula $(\boldsymbol{Q}^\alpha \boldsymbol{A}\boldsymbol{B})$ (expressing equality of two formulae $\boldsymbol{A}$ and $\boldsymbol{B}$ of type α) β-reduces to $\forall P_{\alpha\to o} . (P\boldsymbol{A}) \Rightarrow (P\boldsymbol{B})$, which can be read as: formulae $\boldsymbol{A}$ and $\boldsymbol{B}$ are not equal iff there exists a discerning property P.[2] In other words, $\boldsymbol{A}$ and $\boldsymbol{B}$ are equal, if they are indiscernible. We will use the notation $\boldsymbol{A} \doteq^\alpha \boldsymbol{B}$ as shorthand for the β-reduct $\forall P_{\alpha\to o} . (P\boldsymbol{A}) \Rightarrow (P\boldsymbol{B})$ of $(\boldsymbol{Q}^\alpha \boldsymbol{A}\boldsymbol{B})$ (where $P \notin \mathrm{free}(\boldsymbol{A}) \cup \mathrm{free}(\boldsymbol{B})$).[3] There are alternative ways to define equality in terms of the logical connectives ([6, p. 203]) and the techniques for equality introduced in this article carry over to them (cf. Remark 4.4).

In this article we use several different notions of equality. In order to prevent misunderstandings we explain these different notions together with their syntactical representation here:

If we *define* a concept we use $:=$ (e.g., let $\mathscr{D} := \{\mathtt{T}, \mathtt{F}\}$). $\equiv$ represents identity. We refer to a representative of the identity relation on $\mathscr{D}_\alpha$ as an *object* of the *semantical domain* $\mathscr{D}_{\alpha\to\alpha\to o}$ with q^α. Note that we possibly have one, several, or no q^α in $\mathscr{D}_{\alpha\to\alpha\to o}$ for each domain $\mathscr{D}_\alpha$. The remaining two notions are related to syntax. $=^\alpha$ may occur as a *constant symbol* of type $\alpha \to \alpha \to o$ in a signature Σ. Finally, $\doteq^\alpha$ and $\mathbf{Q}^\alpha$ are used for *Leibniz equality* as described above.

2.3. Notions of models for $\mathscr{HOL}$. A model of $\mathscr{HOL}$ is a collection of non-empty domains $\mathscr{D}_\alpha$ for all types α together with a way of interpreting formulae. The model classes discussed in this article will vary in the domains and specifics of the evaluation of formulae. The relationships between these classes of models are depicted as a cube in Figure 1. We will discuss the model classes from bottom to top, from the most specific notion of standard models ($\mathfrak{ST}$) to the most general notion of υ-complexes, motivating the respective generalizations as we go along. In Section 3, where we develop the theory formally based on the intuitions discussed

[2]Note that this is symmetric by considering complements and hence it is sufficient to use $\Rightarrow$ instead of $\Leftrightarrow$.

[3]Note that $\boldsymbol{A} \doteq^\alpha \boldsymbol{B}$ is β-normal iff $\boldsymbol{A}$ and $\boldsymbol{B}$ are β-normal. The same holds for $\beta\eta$-equality.

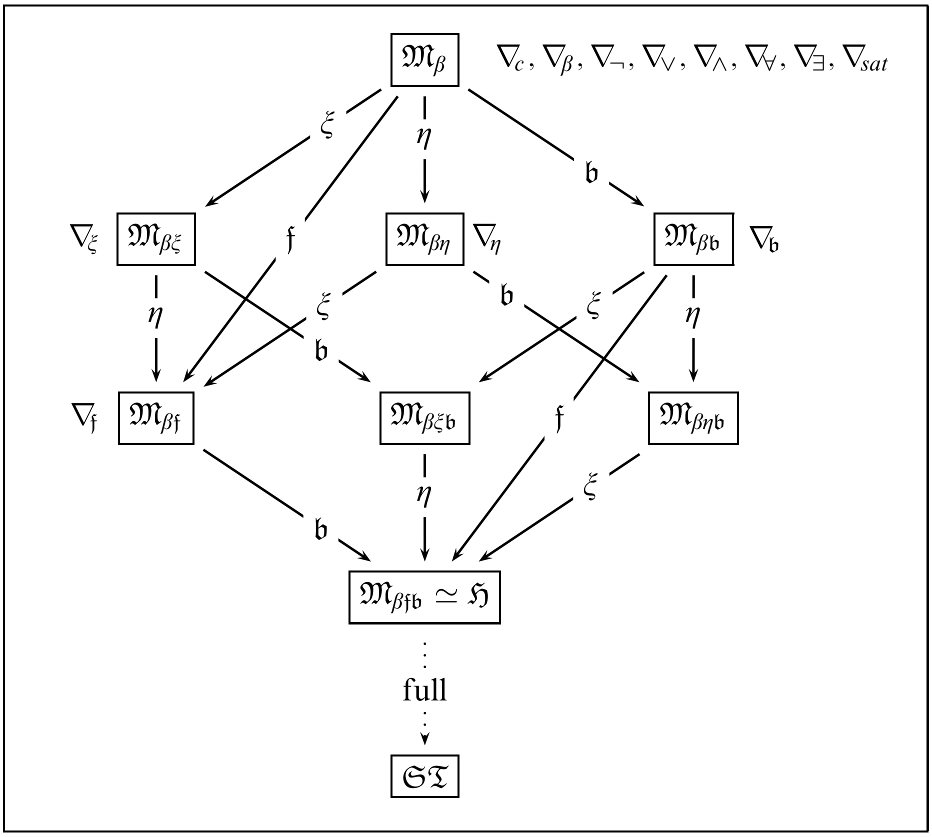

FIGURE 1. The landscape of higher-order semantics.

here, we will proceed the other way around, specializing the notion of a Σ-model more and more.

The symbols in the boxes in Figure 1 denote model classes, the symbols labeling the arrows indicate the properties inducing the corresponding specialization, and the ∇-symbols next to the boxes indicate the clauses in the definition of abstract consistency classes (cf. Definition 6.5) that are needed to establish a model existence theorem for this particular class of models (cf. Theorem 6.34).

2.3.1. *Standard and Henkin models* [$\mathfrak{ST}, \mathfrak{H}, \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$]. A *standard model* ($\mathfrak{ST}$, cf. Definition 3.51) for $\mathscr{HOL}$ provides a fixed set $\mathscr{D}_\iota$ of individuals and a set $\mathscr{D}_o := \{\mathtt{T}, \mathtt{F}\}$ of truth values. All the domains for the function types are defined inductively: $\mathscr{D}_{\alpha\to\beta}$ is the set of functions $f : \mathscr{D}_\alpha \longrightarrow \mathscr{D}_\beta$. The evaluation function $\mathscr{E}_\varphi$ with respect to an assignment φ of variables is obtained by the standard homomorphic construction that evaluates a λ-abstraction with a function.

One can reconstruct the key idea behind *Henkin models* ($\mathfrak{H}$ isomorphic to $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$, cf. Definitions 3.50, and Theorem 3.68) by the following observation. If the set $\mathscr{D}_\iota$ is infinite, the set $\mathscr{D}_{\iota\to o}$ of sets of individuals must be uncountably infinite. On the other hand, any reasonable semantics of a language with a countable signature that admits sound and complete calculi must have countable models. Leon Henkin generalized the class of admissible domains for functional types [26]. Instead of requiring $\mathcal{D}_{\alpha\rightarrow\beta}$ (and thus in particular, $\mathcal{D}_{\iota\rightarrow o}$) to be the full set of functions (predicates), it is sufficient to require that $\mathcal{D}_{\alpha\rightarrow\beta}$ has enough members that any well-formed formula can be evaluated (in other words, the domains of function types are rich enough to satisfy comprehension). Note that with this generalized notion of a model, there are fewer formulae that are valid in all models (intuitively, for any given formula there are more possibilities for counter-models). The generalization to Henkin models restricts the set of valid formulae sufficiently so that all of them can be proven by a Hilbert-style calculus [26].

Of course our picture in Figure 1 is not complete here; we can axiomatically require the existence of particular (classes of) functions, e.g., by assuming the description or choice operators. We will not pursue this here; for a detailed discussion of the semantic issues raised by the presence of these logical constants see [3]. Note that even though we can consider model classes with richer and richer function spaces, we can never reach standard models where function spaces are full while maintaining complete (recursively axiomatizable) calculi.

2.3.2. *Models without boolean extensionality* $[\mathfrak{M}_\beta, \mathfrak{M}_{\beta\xi}, \mathfrak{M}_{\beta\eta}, \mathfrak{M}_{\beta\mathfrak{f}}]$. The next generalization of model classes comes from the fact that we want to have logics where the axiom of Boolean extensionality can fail. For instance, in the semantics of natural language we have so-called verbs and adjectives of "propositional attitude" like *believe* or *obvious*. We may not want to commit ourselves to a logic where the sentence "John believes that Phil is a woodchuck" automatically entails "John believes that Phil is a groundhog" since John might not be aware that "woodchuck" is just another word for "groundhog". The axiom of Boolean extensionality does just that; it states that whenever two propositions are equivalent, they must be equal, and can be substituted for each other. Similarly, the formulae obvious($\boldsymbol{O}$) and obvious($\boldsymbol{F}$) where $\boldsymbol{O} := 2 + 2 = 4$ and $\boldsymbol{F} := \forall n > 2.x^n + y^n = z^n \Rightarrow x = y = z = 0$ should not be equivalent, even if their arguments are. (Both $\boldsymbol{O}$ and $\boldsymbol{F}$ are true over the natural numbers, but Fermat's last theorem $\boldsymbol{F}$ is non-obvious to most people). These phenomena have been studied under the heading of "hyper-intensional semantics" in theoretical semantics; see [39] for a survey.

To account for this behavior, we have to generalize the class of Henkin models further so that there are counter-models to the examples above. Obviously, this involves weakening the assumption that $\mathcal{D}_o \equiv \{\mathtt{T}, \mathtt{F}\}$ since this entails that the values of $\boldsymbol{O}$ and $\boldsymbol{F}$ are identical. We call the assumption that $\mathcal{D}_o$ has two elements property $\mathfrak{b}$. In our Σ-*models* without property $\mathfrak{b}$ ($\mathfrak{M}_\beta$, $\mathfrak{M}_{\beta\xi}$, $\mathfrak{M}_{\beta\eta}$, $\mathfrak{M}_{\beta\mathfrak{f}}$, cf. Definitions 3.41 and 3.49) we only insist that there is a division of the truth values into "good" and "bad" ones, which we express by insisting on the existence of a valuation υ of $\mathcal{D}_o$, i.e., a function $\upsilon\colon \mathcal{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ that is coordinated with the interpretations of the logical constants $\neg$, $\vee$, and Π^α (for each type α). Thus we have a notion of validity: we call a sentence $\boldsymbol{A}$ valid in such a model if $\upsilon(\mathsf{a}) \equiv \mathtt{T}$, where $\mathsf{a} \in \mathcal{D}_o$ is the value of the sentence $\boldsymbol{A}$. For example, there is a Σ-model (see Examples 5.4 and 5.5) where woodchuck(phil), groundhog(phil) and believe(john, woodchuck(phil)) are all valid, but believe(john, groundhog(phil)) is not. In this model, the value of woodchuck(phil) is different from the value of groundhog(phil) in $\mathcal{D}_o$.

2.3.3. *Models without functional extensionality* [$\mathfrak{M}_\beta$, $\mathfrak{M}_{\beta\eta}$, $\mathfrak{M}_{\beta\xi}$, $\mathfrak{M}_{\beta\mathfrak{b}}$, $\mathfrak{M}_{\beta\eta\mathfrak{b}}$, $\mathfrak{M}_{\beta\xi\mathfrak{b}}$]. In mathematics (and as a consequence in most higher-order model theories), we assume functional extensionality, which states that two functions are equal, if they return identical values on all arguments. In many applications we want to use a logic that allows a finer-grained modeling of properties of functions. For instance, if we want to model programs as (higher-order) functions, we might be interested in intensional[4] properties like run-time complexity. Consider for instance the two functions $\boldsymbol{I} := \lambda X.X$ and $\boldsymbol{L} := \lambda X.\mathrm{rev}(\mathrm{rev}(X))$, where rev is the self-inverse function that reverses the order of elements in a list. While the identity function has constant complexity, the function rev is linear in the length of its argument. As a consequence, even though $\boldsymbol{L}$ behaves like $\boldsymbol{I}$ on all inputs, they have different time complexity. A logic with a functionally extensional model theory (which is encoded as property $\mathfrak{f}$, cf. Definitions 3.5, 3.41 and 3.46) would conflate $\boldsymbol{I}$ and $\boldsymbol{L}$ semantically and thus hide this difference rendering the logic unsuitable for complexity analysis.

To arrive at a model theory which does not require functional extensionality (which we will a call non-functional model theory in the remainder) we need to generalize the notion of domains at function types and evaluation functions. This is because the usual construction already uses sets of (extensional) functions for the domains of function type and the property of functionality to construct values for λ-terms.

We build on the notion of applicative structures (cf. Definition 3.1) to define Σ-evaluations (cf. Definition 3.18), where the evaluation function is assumed to respect application and β-conversion. In such models, a function is not uniquely determined by its behavior on all possible arguments. Such models can be constructed, for example, by labeling for functions (e.g., a green and a red version of a function f) in order to differentiate between them, even though they are functionally equivalent (cf. Example 5.6). Property $\mathfrak{b}$ may or may not hold for non-functional Σ-Models.

We can factor functional extensionality (property $\mathfrak{f}$) into two independent properties, property η and property ξ. A model satisfies property η if it respects η-conversion. A model satisfies property ξ if we can conclude the values of $\lambda X.\boldsymbol{M}$ and $\lambda X.\boldsymbol{N}$ are identical whenever the values of $\boldsymbol{M}$ and $\boldsymbol{N}$ are identical for any assignment of the variable X. We will show that a model satisfies property $\mathfrak{f}$ iff it satisfies both property η and property ξ (cf. Lemma 3.24).

2.3.4. *Andrews' models and υ-complexes* [$\mathfrak{M}_\beta, \mathfrak{M}_{\beta\eta}$]. Peter Andrews has pioneered the construction of non-functional models with his υ-complexes in [1] based on Kurt Schütte's semi-valuation method [50]. These constructions, where both functional and Boolean extensionality fail, are Σ-models as defined in Definition 3.41. (Typically they will not even satisfy the property that Leibniz equality corresponds to identity in the model, but they will have a quotient by Theorem 3.62 which does satisfy this property.)

2.4. Characterizing the deductive power of calculi. These model classes discussed in the previous section characterize the deductive power of many higher-order

[4] Just as in the linguistic application, the word "intensional" is used as a synonym for "non-extensional" even though totally different properties are intended.

theorem provers on a semantic level. For example, TPS [8] can be used in modes in which the deductive power is characterized by $\mathfrak{M}_{\beta\eta}$ (or even $\mathfrak{M}_{\beta}$ if η-conversion is disallowed). Note that in particular TPS is not complete with respect to Henkin models. It is not even complete for $\mathfrak{M}_{\beta\eta\mathfrak{b}}$, although it can be used in modes with some 'extensionality treatment' built into the proof procedure.

The incompleteness of TPS for Henkin models[5] can be seen from the fact that it fails to refute formulae such as $c\boldsymbol{A}_o \wedge \neg c(\neg\neg\boldsymbol{A})$, where c is a constant of type $o \rightarrow o$, or to prove formulae like $p(\lambda X_\alpha \boldsymbol{.} \boldsymbol{B}X \wedge \boldsymbol{A}X) \Rightarrow p(\lambda X_\alpha \boldsymbol{.} \boldsymbol{A}X \wedge \boldsymbol{B}X)$, where p is a constant of type $(\alpha \rightarrow o) \rightarrow o$. The problem in the former example is that the higher-order unification algorithm employed by TPS cannot determine that $\boldsymbol{A}$ and $\neg\neg\boldsymbol{A}$ denote identical semantic objects (by Boolean extensionality as already mentioned before), and thus returns failure instead of success. In the second example both functional and Boolean extensionality are needed in order to prove the theorem.

[21] discusses a presentation of higher-order logic in a first-order logic based on an approach called *theorem proving modulo*. It is easy to check that this approach is also incomplete for model classes with property $\mathfrak{b}$. For instance the approach cannot prove the formula

$$\forall P_{o\rightarrow o} X_o Y_o \boldsymbol{.} (PX \wedge PY) \Rightarrow P(X \wedge Y)$$

which is valid in Henkin models and which requires $\mathfrak{b}$. As a result, the *theorem proving modulo* approach of representing higher-order logic in a first-order logic [21] can only be used for logics without Boolean extensionality in its current form.

2.4.1. *Model existence theorems.* For all the notions of model classes (except, of course, for standard models, where such a theorem cannot hold for recursively axiomatizable logical systems) we present model existence theorems tying the differentiating conditions of the models to suitable conditions in the abstract consistency classes (cf. Section 6.3).

A model existence theorem for a logical system $\mathcal{S}$ (i.e., a logical language $\mathcal{L}_\mathcal{S}$ together with a consequence relation $\models_\mathcal{S} \subseteq \mathcal{L}_\mathcal{S} \times \mathcal{L}_\mathcal{S}$) is a theorem of the form:

> *If a set of sentences Φ of $\mathcal{S}$ is a member of an abstract consistency class Γ, then there exists a $\mathcal{S}$-model for Φ.*

For the proof we can use the classical construction in all cases: abstract consistent sets are extended to Hintikka sets (cf. Section 6.2), which induce a valuation on a term structure (cf Definition 3.35). We then take a quotient by the congruence induced by Leibniz equality in the term model.

2.4.2. *Completeness of calculi.* Given a model existence theorem as described above we can show the completeness of a particular calculus $\mathscr{C}$ (i.e., the derivability relation $\vdash_\mathcal{S} \subseteq \mathcal{L}_\mathcal{S} \times \mathcal{L}_\mathcal{S}$) by proving that the class Γ of sets of sentences Φ that are $\mathscr{C}$-consistent (i.e., cannot be refuted in $\mathscr{C}$) is an abstract consistency class. Then the model existence theorem tells us that $\mathscr{C}$-consistent sets of sentences are satisfiable in $\mathcal{S}$. Now we assume that a sentence $\boldsymbol{A}$ is valid in $\mathcal{S}$, so $\neg\boldsymbol{A}$ does not have a $\mathcal{S}$-model and is therefore $\mathscr{C}$-inconsistent. Hence, $\neg\boldsymbol{A}$ is refutable in $\mathscr{C}$. This shows

[5]In case the extensionality axioms are not available in the search space. Note that one can add extensionality axioms to the calculus in order to achieve—at least in theory—Henkin completeness. But this increases the search space drastically and is not feasible in practice.

refutation completeness of $\mathscr{C}$. For many calculi $\mathscr{C}$, this also shows $\boldsymbol{A}$ is provable, thus establishing completeness of $\mathscr{C}$.

Note that with this argumentation the completeness proof for $\mathscr{C}$ condenses to verifying that Γ is an abstract consistency class, a task that does not refer to $\mathscr{S}$-models. Thus the usefulness of model existence theorems derives from the fact that it replaces the model-theoretic analysis in completeness proofs with the verification of some proof-theoretic conditions. In this respect a model existence theorem is similar to a Herbrand Theorem, but it is easier to generalize to other logic systems like higher-order logic. The technique was developed for first-order logic by Jaakko Hintikka and Raymond Smullyan [29, 52, 53].

§3. Semantics for higher-order logic.

In this section we will introduce the semantical constructions and discuss their relationships. We will start out by defining applicative structures and Σ-evaluations to give an algebraic semantics for the simply typed λ-calculus. To obtain a model for higher-order logic, we use a Σ-valuation to determine whether propositions are true or false.

3.1. Applicative structures.

DEFINITION 3.1 ((Typed) Applicative structure). A collection $\mathscr{D} := \mathscr{D}_{\mathscr{T}} := \{\, \mathscr{D}_\alpha \mid \alpha \in \mathscr{T} \,\}$ of non-empty sets $\mathscr{D}_\alpha$, indexed by the set $\mathscr{T}$ of types, is called a *typed collection* (of sets). Let $\mathscr{D}_{\mathscr{T}}$ and $\mathscr{E}_{\mathscr{T}}$ be typed collections, then a collection $\mathsf{f} := \{\, \mathsf{f}^\alpha \colon \mathscr{D}_\alpha \longrightarrow \mathscr{E}_\alpha \mid \alpha \in \mathscr{T} \,\}$ of functions is called a *typed function* $\mathsf{f} \colon \mathscr{D}_{\mathscr{T}} \longrightarrow \mathscr{E}_{\mathscr{T}}$. We will write $\mathscr{F}(A; B)$ for the set of functions from A to B and $\mathscr{F}_{\mathscr{T}}(\mathscr{D}_{\mathscr{T}}; \mathscr{E}_{\mathscr{T}})$ for the set of typed functions. In the following we will also use the notion of a typed function extended to the n-ary case in the obvious way.

We call the pair $(\mathscr{D}, @)$ a (typed) *applicative structure* if $\mathscr{D} \equiv \mathscr{D}_{\mathscr{T}}$ is a typed collection of sets and

$$@ := \{\, @^{\alpha\beta} \colon \mathscr{D}_{\alpha\to\beta} \times \mathscr{D}_\alpha \longrightarrow \mathscr{D}_\beta \mid \alpha, \beta \in \mathscr{T} \,\}.$$

Each (non-empty) set $\mathscr{D}_\alpha$ is called the *domain* of type α and the family of functions $@$ is called the *application operator*. We write simply $\mathsf{f}@\mathsf{a}$ for $\mathsf{f}@^{\alpha\beta}\mathsf{a}$ when $\mathsf{f} \in \mathscr{D}_{\alpha\to\beta}$ and $\mathsf{a} \in \mathscr{D}_\alpha$ are clear in context.

REMARK 3.2. Often an applicative structure is defined to also include an interpretation of the constants in a given signature (for example, in [44]). We prefer this signature-independent definition (as in [30]) for our purposes.

REMARK 3.3 (Currying). The application operator $@$ in an applicative structure is an abstract version of function application. It is no restriction to exclusively use a binary application operator, which corresponds to unary function application, since we can define higher-arity application operators from the binary one by setting $\mathsf{f}@(\mathsf{a}^1, \dots, \mathsf{a}^n) := (\dots(\mathsf{f}@\mathsf{a}^1)\dots@\mathsf{a}^n)$ ("Currying").

DEFINITION 3.4 (Frame). An applicative structure $(\mathscr{D}, @)$ is called a *frame*, if $\mathscr{D}_{\alpha\to\beta} \subseteq \mathscr{F}(\mathscr{D}_\alpha; \mathscr{D}_\beta)$ and $@^{\alpha\beta}$ is application for functions for all types α and β.

DEFINITION 3.5 (Functional/full/standard applicative structures). Let $\mathscr{A} := (\mathscr{D}, @)$ be an applicative structure. We say that $\mathscr{A}$ is *functional* if for all types α and β and objects $\mathsf{f}, \mathsf{g} \in \mathscr{D}_{\alpha\to\beta}$, we have $\mathsf{f} \equiv \mathsf{g}$ whenever $\mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a}$ for every

$\mathsf{a} \in \mathscr{D}_\alpha$.[6] We say $\mathscr{A}$ is *full* if for all types α and β and every function $f : \mathscr{D}_\alpha \longrightarrow \mathscr{D}_\beta$ there is an object $\mathsf{f} \in \mathscr{D}_{\alpha\to\beta}$ such that $\mathsf{f}@\mathsf{a} \equiv f(\mathsf{a})$ for every $\mathsf{a} \in \mathscr{D}_\alpha$. Finally, we say $\mathscr{A}$ is *standard* if it is a frame and $\mathscr{D}_{\alpha\to\beta} \equiv \mathscr{F}(\mathscr{D}_\alpha; \mathscr{D}_\beta)$ for all types α and β. Note that these definitions impose restrictions on the domains for function types only.

REMARK 3.6. It is easy to show that every frame is functional. Furthermore, an applicative structure is standard iff it is a full frame.

EXAMPLE 3.7 (Applicative singleton structure). We choose a single element a and define $\mathscr{D}_\alpha := \{\mathsf{a}\}$ for all types α. The pair $(\mathscr{D}_\mathscr{T}, @^{\mathsf{a}})$, where $\mathsf{a}@^{\mathsf{a}}\mathsf{a} = \mathsf{a}$ is a (trivial) example of a functional applicative structure. It is called the *singleton applicative structure*.

EXAMPLE 3.8 (Applicative term structures). If we define $\boldsymbol{A}@\boldsymbol{B} := (\boldsymbol{AB})$ for $\boldsymbol{A} \in \mathrm{wff}_{\alpha\to\beta}(\Sigma)$ and $\boldsymbol{B} \in \mathrm{wff}_\alpha(\Sigma)$, then $@\colon \mathrm{wff}_{\alpha\to\beta}(\Sigma) \times \mathrm{wff}_\alpha(\Sigma) \longrightarrow \mathrm{wff}_\beta(\Sigma)$ is a total function. Thus $(\mathrm{wff}(\Sigma), @)$ is an applicative structure. The intuition behind this example is that we can think of the formula $\boldsymbol{A} \in \mathrm{wff}_{\alpha\to\beta}(\Sigma)$ as a function $\boldsymbol{A}\colon \mathrm{wff}_\alpha(\Sigma) \longrightarrow \mathrm{wff}_\beta(\Sigma)$ that maps $\boldsymbol{B}$ to $(\boldsymbol{AB})$.

Analogously, we can define the applicative structure $(\mathrm{cwff}(\Sigma), @)$ of closed formulae (when we ensure Σ contains enough constants so that $\mathrm{cwff}_\alpha(\Sigma)$ is non-empty for all types α).

DEFINITION 3.9 (Homomorphism). Let $\mathscr{A}^1 := (\mathscr{D}^1, @^1)$ and $\mathscr{A}^2 := (\mathscr{D}^2, @^2)$ be applicative structures. A *homomorphism* from $\mathscr{A}^1$ to $\mathscr{A}^2$ is a typed function $\kappa\colon \mathscr{D}^1 \longrightarrow \mathscr{D}^2$ such that for all types $\alpha, \beta \in \mathscr{T}$, all $\mathsf{f} \in \mathscr{D}^1_{\alpha\to\beta}$, and $\mathsf{a} \in \mathscr{D}^1_\alpha$ we have $\kappa(\mathsf{f})@^2\kappa(\mathsf{a}) \equiv \kappa(\mathsf{f}@^1\mathsf{a})$. We write $\kappa\colon \mathscr{A}^1 \longrightarrow \mathscr{A}^2$. The two applicative structures $\mathscr{A}^1$ and $\mathscr{A}^2$ are called *isomorphic* if there are homomorphisms $i : \mathscr{A}^1 \longrightarrow \mathscr{A}^2$ and $j : \mathscr{A}^2 \longrightarrow \mathscr{A}^1$ which are mutually inverse at each type.

The most important method for constructing structures (and models) with given properties in this article is well-known for algebraic structures and consists of building a suitable congruence and passing to the quotient structure. We will now develop the formal basis for it.

DEFINITION 3.10 (Applicative structure congruences). Let $\mathscr{A} := (\mathscr{D}, @)$ be an applicative structure. A typed equivalence relation $\sim$ is called a *congruence* on $\mathscr{A}$ iff for all $\mathsf{f}, \mathsf{f}' \in \mathscr{D}_{\alpha\to\beta}$ and $\mathsf{a}, \mathsf{a}' \in \mathscr{D}_\alpha$ (for any types α and β), $\mathsf{f} \sim \mathsf{f}'$ and $\mathsf{a} \sim \mathsf{a}'$ imply $\mathsf{f}@\mathsf{a} \sim \mathsf{f}'@\mathsf{a}'$.

The *equivalence class* $[\![\mathsf{a}]\!]_\sim$ of $\mathsf{a} \in \mathscr{D}_\alpha$ *modulo* $\sim$ is the set of all $\mathsf{a}' \in \mathscr{D}_\alpha$, such that $\mathsf{a} \sim \mathsf{a}'$. A congruence $\sim$ is called *functional* iff for all types α and β and $\mathsf{f}, \mathsf{g} \in \mathscr{D}_{\alpha\to\beta}$, we have $\mathsf{f} \sim \mathsf{g}$ whenever $\mathsf{f}@\mathsf{a} \sim \mathsf{g}@\mathsf{a}$ for every $\mathsf{a} \in \mathscr{D}_\alpha$.

LEMMA 3.11. *The β-equality and $\beta\eta$-equality relations $\equiv_\beta$ and $\equiv_{\beta\eta}$ are congruences on the applicative structures* $\mathrm{wff}(\Sigma)$ *and* cwff.

PROOF. The congruence properties are a direct consequence of the fact that $\beta\eta$-reduction rules are defined to act on subterm positions. ⊣

[6] This is called "extensional" in [44]. We use the term "functional" to distinguish it from other forms of extensionality.

DEFINITION 3.12 (Quotient applicative structure). Let $\mathscr{A} := (\mathscr{D}, @)$ be an applicative structure, $\sim$ a congruence on $\mathscr{A}$, and $\mathscr{D}^{\sim}_{\alpha} := \{[\![\mathsf{a}]\!]_{\sim} \mid \mathsf{a} \in \mathscr{D}_{\alpha}\}$. Furthermore, let $@^{\sim}$ be defined by $[\![\mathsf{f}]\!]_{\sim} @^{\sim} [\![\mathsf{a}]\!]_{\sim} := [\![\mathsf{f}@\mathsf{a}]\!]_{\sim}$. (To see that this definition only depends on equivalence classes of $\sim$, consider $\mathsf{f}' \in [\![\mathsf{f}]\!]_{\sim}$ and $\mathsf{a}' \in [\![\mathsf{a}]\!]_{\sim}$. Then $\mathsf{f} \sim \mathsf{f}'$ and $\mathsf{a} \sim \mathsf{a}'$ imply $\mathsf{f}@\mathsf{a} \sim \mathsf{f}'@\mathsf{a}'$. Thus, $[\![\mathsf{f}@\mathsf{a}]\!]_{\sim} \equiv [\![\mathsf{f}'@\mathsf{a}']\!]_{\sim}$. So, $@^{\sim}$ is well-defined.) $\mathscr{A}/_{\sim} := (\mathscr{D}^{\sim}, @^{\sim})$ is also an applicative structure. We call $\mathscr{A}/_{\sim}$ the *quotient structure* of $\mathscr{A}$ for the relation $\sim$ and the typed function $\pi_{\sim} : \mathscr{A} \longrightarrow \mathscr{A}/_{\sim}$ that maps a to $[\![\mathsf{a}]\!]_{\sim}$ its *canonical projection*.

THEOREM 3.13. *Let $\mathscr{A}$ be an applicative structure and let $\sim$ be a congruence on $\mathscr{A}$, then the canonical projection $\pi_{\sim}$ is a surjective homomorphism. Furthermore, $\mathscr{A}/_{\sim}$ is functional iff $\sim$ is functional.*

PROOF. Let $\mathscr{A} := (\mathscr{D}, @)$ be an applicative structure. To convince ourselves that $\pi_{\sim}$ is indeed a surjective homomorphism, we note that $\pi_{\sim}$ is surjective by the definition of $\mathscr{D}^{\sim}$. To see that $\pi_{\sim}$ is a homomorphism let $\mathsf{f} \in \mathscr{D}_{\alpha\to\beta}$, and $\mathsf{a} \in \mathscr{D}_{\beta}$, then $\pi_{\sim}(\mathsf{f}) @^{\sim} \pi_{\sim}(\mathsf{a}) \equiv [\![\mathsf{f}]\!]_{\sim} @^{\sim} [\![\mathsf{a}]\!]_{\sim} \equiv [\![\mathsf{f}@\mathsf{a}]\!]_{\sim} \equiv \pi_{\sim}(\mathsf{f}@\mathsf{a})$.

The quotient construction collapses $\sim$ to identity, so functionality of $\sim$ is equivalent to functionality of $\mathscr{A}/_{\sim}$. Formally, suppose $[\![\mathsf{f}]\!]_{\sim}$ and $[\![\mathsf{g}]\!]_{\sim}$ are elements of $\mathscr{D}^{\sim}_{\alpha\to\beta}$ such that $[\![\mathsf{f}]\!]_{\sim} @^{\sim} [\![\mathsf{a}]\!]_{\sim} \equiv [\![\mathsf{g}]\!]_{\sim} @^{\sim} [\![\mathsf{a}]\!]_{\sim}$ for every $[\![\mathsf{a}]\!]_{\sim}$ in $\mathscr{D}^{\sim}_{\alpha}$. This is equivalent to $[\![\mathsf{f}@\mathsf{a}]\!]_{\sim} \equiv [\![\mathsf{g}@\mathsf{a}]\!]_{\sim}$ for every $\mathsf{a} \in \mathscr{D}_{\alpha}$ and hence $\mathsf{f}@\mathsf{a} \sim \mathsf{g}@\mathsf{a}$ for all $\mathsf{a} \in \mathscr{D}_{\alpha}$. By functionality of $\sim$, we have $\mathsf{f} \sim \mathsf{g}$. That is, $[\![\mathsf{f}]\!]_{\sim} \equiv [\![\mathsf{g}]\!]_{\sim}$. ⊣

LEMMA 3.14. *$\equiv_{\beta\eta}$ is a functional congruence on* $\mathrm{wff}(\Sigma)$. *If Σ_{α} is infinite for all types $\alpha \in \mathscr{T}$, then $\equiv_{\beta\eta}$ is also functional on* cwff.

PROOF. By Lemma 3.11, $\equiv_{\beta\eta}$ is a congruence relation. To show functionality let $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_{\gamma\to\alpha}(\Sigma)$ such that $\boldsymbol{AC} \equiv_{\beta\eta} \boldsymbol{BC}$ for all $\boldsymbol{C} \in \mathrm{wff}_{\gamma}(\Sigma)$ be given. In particular, for any variable $X \in \mathscr{V}_{\gamma}$ that is not free in $\boldsymbol{A}$ or $\boldsymbol{B}$, we have $\boldsymbol{A}X \equiv_{\beta\eta} \boldsymbol{B}X$ and $\lambda X.\boldsymbol{A}X \equiv_{\beta\eta} \lambda X.\boldsymbol{B}X$. By definition we have $\boldsymbol{A} \equiv_{\eta} \lambda X_{\gamma}.\boldsymbol{A}X \equiv_{\beta\eta} \lambda X_{\gamma}.\boldsymbol{B}X \equiv_{\eta} \boldsymbol{B}$.

To show functionality of $\beta\eta$-equality on closed formulae, suppose $\boldsymbol{A}$ and $\boldsymbol{B}$ are closed. With the same variable X as above, let $\boldsymbol{M}$ and $\boldsymbol{N}$ be the $\beta\eta$-normal forms of $\boldsymbol{A}X$ and $\boldsymbol{B}X$, respectively. We cannot conclude that $\boldsymbol{M} \equiv \boldsymbol{N}$ since X is not a closed term. Instead, choose a constant $c_{\gamma} \in \Sigma_{\gamma}$ that does not occur in $\boldsymbol{A}$ or $\boldsymbol{B}$. (Such a constant must exist, since we have assumed that Σ_{γ} is infinite.) An easy induction on the length of the $\beta\eta$-reduction sequence from $\boldsymbol{A}X$ to $\boldsymbol{M}$ shows that c does not occur in $\boldsymbol{M}$ and $\boldsymbol{A}c \equiv [c/X](\boldsymbol{A}X)$ $\beta\eta$-reduces to $[c/X]\boldsymbol{M}$. Similarly, c does not occur in $\boldsymbol{N}$ and $\boldsymbol{B}c$ $\beta\eta$-reduces to $[c/X]\boldsymbol{N}$. Since c is a constant, substituting c for X cannot introduce new redexes. So, simple inductions on the sizes of $\boldsymbol{M}$ and $\boldsymbol{N}$ show $[c/X]\boldsymbol{M}$ and $[c/X]\boldsymbol{N}$ are $\beta\eta$-normal. By assumption, we know $\boldsymbol{A}c \equiv_{\beta\eta} \boldsymbol{B}c$. Since normal forms are unique, we must have $[c/X]\boldsymbol{M} \equiv [c/X]\boldsymbol{N}$. Using the fact that c does not occur in either $\boldsymbol{M}$ or $\boldsymbol{N}$, an induction on the size of $\boldsymbol{M}$ readily shows $\boldsymbol{M} \equiv \boldsymbol{N}$. So, we have $\boldsymbol{A} \equiv_{\eta} \lambda X_{\gamma}.\boldsymbol{A}X \equiv_{\beta\eta} \lambda X_{\gamma}.\boldsymbol{M} \equiv \lambda X_{\gamma}.\boldsymbol{N} \equiv_{\beta\eta} \lambda X_{\gamma}.\boldsymbol{B}X \equiv_{\eta} \boldsymbol{B}$ ⊣

REMARK 3.15. Suppose we have a signature Σ with a single constant c_{ι}. In this case, c is the only closed $\beta\eta$-normal form of type ι. Since $\lambda X.X \not\equiv_{\beta\eta} \lambda X.c$ even though $(\lambda X.X)c \equiv_{\beta\eta} c \equiv_{\beta\eta} (\lambda X.c)c$ we have a counterexample to functionality of $\equiv_{\beta\eta}$ on cwff. The problem here is that we do not have another constant d_{ι} to distinguish the two functions. In $\mathrm{wff}(\Sigma)$ we could always use a variable.

REMARK 3.16 (Assumptions on Σ). From now on, we assume Σ_α to be infinite for each type α. Furthermore, we assume there is a particular cardinal $\aleph_s$ such that Σ_α has cardinality $\aleph_s$ for every type α. Since $\mathscr{V}$ is countable, this implies $\mathrm{wff}_\alpha(\Sigma)$ and cwff_α have cardinality $\aleph_s$ for each type α. Also, whether or not primitive equality is included in the signature, there can only be finitely many logical constants in Σ_α for each particular type α. Thus, the cardinality of the set of parameters in Σ_α is also $\aleph_s$. In the countable case, $\aleph_s$ is $\aleph_0$.

3.2. Σ-evaluations. Σ-evaluations are applicative structures with a notion of evaluation for well-formed formulae in $\mathrm{wff}(\Sigma)$.

DEFINITION 3.17 (Variable assignment). Let $\mathscr{A} := (\mathscr{D}, @)$ be an applicative structure. A typed function $\varphi\colon \mathscr{V} \longrightarrow \mathscr{D}$ is called a *variable assignment* into $\mathscr{A}$. Given a variable assignment φ, variable X_α, and value $\mathsf{a} \in \mathscr{D}_\alpha$, we use $\varphi, [\mathsf{a}/X]$ to denote the variable assignment with $(\varphi, [\mathsf{a}/X])(X) \equiv \mathsf{a}$ and $(\varphi, [\mathsf{a}/X])(Y) \equiv \varphi(Y)$ for variables Y other than X.

DEFINITION 3.18 (Σ-evaluation). Let $\mathscr{E}\colon \mathscr{F}_{\mathscr{T}}(\mathscr{V}; \mathscr{D}) \longrightarrow \mathscr{F}_{\mathscr{T}}(\mathrm{wff}(\Sigma), \mathscr{D})$ be a total function, where $\mathscr{F}_{\mathscr{T}}(\mathscr{V}; \mathscr{D})$ is the set of variable assignments and $\mathscr{F}_{\mathscr{T}}(\mathrm{wff}(\Sigma), \mathscr{D})$ is the set of typed functions mapping terms into objects in $\mathscr{D}$. We will write the argument of $\mathscr{E}$ as a subscript. So, for each assignment φ, we have a typed function $\mathscr{E}_\varphi\colon \mathrm{wff}(\Sigma) \longrightarrow \mathscr{D}$. $\mathscr{E}$ is called an *evaluation function* for $\mathscr{A}$ if for any assignments φ and ψ into $\mathscr{A}$, we have

(1) $\mathscr{E}_\varphi|_{\mathscr{V}} \equiv \varphi$.
(2) $\mathscr{E}_\varphi(\boldsymbol{FA}) \equiv \mathscr{E}_\varphi(\boldsymbol{F})@\mathscr{E}_\varphi(\boldsymbol{A})$ for any $\boldsymbol{F} \in \mathrm{wff}_{\alpha\to\beta}(\Sigma)$ and $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ and types α and β.
(3) $\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathscr{E}_\psi(\boldsymbol{A})$ for any type α and $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, whenever φ and ψ coincide on $\mathrm{free}(\boldsymbol{A})$.
(4) $\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathscr{E}_\varphi(\boldsymbol{A}{\downarrow_\beta})$ for all $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$.

We call $\mathscr{J} := (\mathscr{D}, @, \mathscr{E})$ a Σ-*evaluation* if $(\mathscr{D}, @)$ is an applicative structure and $\mathscr{E}$ is an evaluation function for $(\mathscr{D}, @)$. We call $\mathscr{E}_\varphi(\boldsymbol{A}_\alpha) \in \mathscr{D}_\alpha$ the *denotation* of $\boldsymbol{A}_\alpha$ in $\mathscr{J}$ for φ. (Note that since $\mathscr{E}$ is a function, the denotation in $\mathscr{J}$ is unique. However, for a given applicative structure $\mathscr{A}$, there may be many possible evaluation functions.)

If $\boldsymbol{A}$ is a closed formula, then $\mathscr{E}_\varphi(\boldsymbol{A})$ is independent of φ, since $\mathrm{free}(\boldsymbol{A}) = \emptyset$. In these cases we sometimes drop the reference to φ from $\mathscr{E}_\varphi(\boldsymbol{A})$ and simply write $\mathscr{E}(\boldsymbol{A})$.

We call a Σ-evaluation $\mathscr{J} := (\mathscr{D}, @, \mathscr{E})$ *functional* [*full*, *standard*] if the applicative structure $(\mathscr{D}, @)$ is *functional* [*full*, *standard*]. We say $\mathscr{J}$ is a Σ-evaluation over a frame if $(\mathscr{D}, @)$ is a frame.

Σ-evaluations generalize Σ-evaluations over frames, which are the basis for Henkin models, to the non-functional case. The existence of an evaluation function that meets the conditions above seems to be the weakest situation where one would like to speak of a model. We cannot in general assume the evaluation function is uniquely determined by its values on constants as this requires functionality. For example, two evaluation functions $\mathscr{E}$ and $\mathscr{E}'$ on the same applicative structure may agree on all constants, but give a different value to the term $(\lambda X_\iota . X)$. Such an example is constructed and discussed later in Remark 5.7.

REMARK 3.19 (Σ-evaluations respect β-equality). Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation and $\boldsymbol{A} \equiv_\beta \boldsymbol{B}$. For all assignments φ into $(\mathcal{D}, @)$, we have $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{A}{\downarrow_\beta}) \equiv \mathcal{E}_\varphi(\boldsymbol{B}{\downarrow_\beta}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$.

We can easily show Σ-evaluations satisfy a *Substitution-Value Lemma*.

LEMMA 3.20 (Substitution-value lemma). *Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation and φ be an assignment into $\mathcal{J}$. For any types α and β, variables X_β, and formulae $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ and $\boldsymbol{B} \in \mathrm{wff}_\beta(\Sigma)$, we have $\mathcal{E}_{\varphi,[\mathcal{E}_\varphi(\boldsymbol{B})/X]}(\boldsymbol{A}) \equiv \mathcal{E}_\varphi([\boldsymbol{B}/X]\boldsymbol{A})$.*

PROOF. Using the fact that $\mathcal{E}$ respects β-equality (cf. Remark 3.19) and the other properties of $\mathcal{E}$ (cf. Definition 3.18), we can compute

$$\begin{aligned}
\mathcal{E}_{\varphi,[\mathcal{E}_\varphi(\boldsymbol{B})/X]}(\boldsymbol{A}) &\equiv \mathcal{E}_{\varphi,[\mathcal{E}_\varphi(\boldsymbol{B})/X]}((\lambda X.\boldsymbol{A})X) \\
&\equiv \mathcal{E}_{\varphi,[\mathcal{E}_\varphi(\boldsymbol{B})/X]}(\lambda X.\boldsymbol{A}) @ \mathcal{E}_{\varphi,[\mathcal{E}_\varphi(\boldsymbol{B})/X]}(X) \\
&\equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{A}) @ \mathcal{E}_\varphi(\boldsymbol{B}) \\
&\equiv \mathcal{E}_\varphi((\lambda X.\boldsymbol{A})\boldsymbol{B}) \\
&\equiv \mathcal{E}_\varphi([\boldsymbol{B}/X]\boldsymbol{A}). \qquad \dashv
\end{aligned}$$

We will consider two weaker notions of functionality. These forms are often discussed in the literature (cf. [28]).

DEFINITION 3.21 (Weakly functional evaluations). Let $\mathcal{J} \equiv (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation. We say $\mathcal{J}$ is *η-functional* if $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{A}{\downarrow_{\beta\eta}})$ for any type α, formula $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, and assignment φ. We say $\mathcal{J}$ is *ξ-functional* if for all $\alpha, \beta \in \mathcal{T}$, $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, assignments φ, and variables X_α, $\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{M}_\beta) \equiv \mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{N}_\beta)$ whenever $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$.

We will now establish that functionality is equivalent to η-functionality and ξ-functionality combined. We prepare for this by first proving two lemmas about functional Σ-evaluations.

LEMMA 3.22. *Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a functional Σ-evaluation. For any assignment φ into $\mathcal{J}$ and $\boldsymbol{F} \in \mathrm{wff}_{\alpha\to\beta}(\Sigma)$ where $X_\alpha \notin \mathrm{free}(\boldsymbol{F})$, we have*

$$\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{F}X) \equiv \mathcal{E}_\varphi(\boldsymbol{F}).$$

PROOF. Let $\mathsf{a} \in \mathcal{D}_\alpha$ be given. Since $X_\alpha \notin \mathrm{free}(\boldsymbol{F})$, we have $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{F}) \equiv \mathcal{E}_\varphi(\boldsymbol{F})$. Since $\mathcal{E}$ respects β-equality (cf. Remark 3.19), we can compute

$$\mathcal{E}_\varphi(\lambda X.\boldsymbol{F}X) @ \mathsf{a} \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}((\lambda X.\boldsymbol{F}X)X) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{F}X) \equiv \mathcal{E}_\varphi(\boldsymbol{F}) @ \mathsf{a}.$$

Generalizing over a, we conclude $\mathcal{E}_\varphi(\lambda X.\boldsymbol{F}X) \equiv \mathcal{E}_\varphi(\boldsymbol{F})$ by functionality. $\dashv$

LEMMA 3.23. *Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a functional Σ-evaluation. If a formula $\boldsymbol{A}$ η-reduces to $\boldsymbol{B}$ in one step, then for any assignment φ into $\mathcal{J}$, $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$.*

PROOF. We prove this by induction on the structure of the term $\boldsymbol{A}$. For the base case when $\boldsymbol{A}$ is the η-redex which is reduced, we apply Lemma 3.22. When $\boldsymbol{A} \equiv (\boldsymbol{FC})$, then the η-reduction either occurs in $\boldsymbol{F}$ or $\boldsymbol{C}$. So, $\boldsymbol{B} \equiv (\boldsymbol{GD})$ where $\boldsymbol{F}$ η-reduces to $\boldsymbol{G}$ in one step (or $\boldsymbol{G} \equiv \boldsymbol{F}$) and $\boldsymbol{D} \equiv \boldsymbol{C}$ (or $\boldsymbol{C}$ η-reduces to $\boldsymbol{D}$ in one step). So, by induction we have $\mathcal{E}_\varphi(\boldsymbol{F}) \equiv \mathcal{E}_\varphi(\boldsymbol{G})$ and $\mathcal{E}_\varphi(\boldsymbol{C}) \equiv \mathcal{E}_\varphi(\boldsymbol{D})$. It follows that $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$.

When $\boldsymbol{A}$ is a λ-abstraction, we must use functionality. Suppose for some type α, $\boldsymbol{A} \equiv (\lambda X_\alpha.\boldsymbol{C})$ (and this is not the η-redex reduced to obtain $\boldsymbol{B}$). Then $\boldsymbol{B} \equiv (\lambda X_\alpha \boldsymbol{D})$

where $\boldsymbol{C}$ η-reduces in one step to $\boldsymbol{D}$. By the induction hypothesis, for any $\mathsf{a} \in \mathcal{D}_\alpha$, $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{C}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{D})$. Since $\mathcal{E}$ is an evaluation function, we have

$$\begin{aligned}\mathcal{E}_\varphi(\lambda X.\boldsymbol{C})@\mathsf{a} &\equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}((\lambda X.\boldsymbol{C})X) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{C})\\ &\equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{D}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}((\lambda X.\boldsymbol{D})X) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{D})@\mathsf{a}.\end{aligned}$$

By functionality, $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{C}) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{D}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$. ⊣

LEMMA 3.24 (Functionality). *Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation. Then $\mathcal{J}$ is functional iff it is both η-functional and ξ-functional.*

PROOF. The fact that functionality implies η-functionality now follows from a simple induction on the number of $\beta\eta$-reduction steps using Lemma 3.23 and Remark 3.19.

To show functionality implies ξ-functionality, let $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, an assignment φ and a variable X_α be given. Suppose $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$. We need to show $\mathcal{E}_\varphi(\lambda X.\boldsymbol{M}) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{N})$. This follows from functionality since

$$\begin{aligned}\mathcal{E}_\varphi(\lambda X.\boldsymbol{M})@\mathsf{a} &\equiv \mathcal{E}_{\varphi,[\mathsf{a},X]}((\lambda X.\boldsymbol{M})X) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M})\\ &\equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N}) \equiv \mathcal{E}_{\varphi,[\mathsf{a},X]}((\lambda X.\boldsymbol{N})X) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{N})@\mathsf{a}\end{aligned}$$

for every $\mathsf{a} \in \mathcal{D}_\alpha$.

To show functionality from η-functionality and ξ-functionality, let $\mathsf{f}, \mathsf{g} \in \mathcal{D}_{\alpha\to\beta}$ such that $\mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a}$ for all $\mathsf{a} \in \mathcal{D}_\alpha$ be given. We need to show that $\mathsf{f} \equiv \mathsf{g}$. Let $F_{\alpha\to\beta}$, $G_{\alpha\to\beta}$ and X_α be variables and φ be any assignment such that $\varphi(F) \equiv \mathsf{f}$ and $\varphi(G) \equiv \mathsf{g}$. Then for any $\mathsf{a} \in \mathcal{D}_\alpha$ we have $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(FX) \equiv \mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a} \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(GX)$, and thus $\mathcal{E}_\varphi(\lambda X.FX) \equiv \mathcal{E}_\varphi(\lambda X.GX)$ by ξ-functionality. Hence,

$$\mathsf{f} \equiv \mathcal{E}_\varphi(F) \equiv \mathcal{E}_\varphi(\lambda X.FX) \equiv \mathcal{E}_\varphi(\lambda X.GX) \equiv \mathcal{E}_\varphi(G) \equiv \mathsf{g}$$

by η-functionality. ⊣

LEMMA 3.25 (ξ-functionality and replacement). *Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a ξ-functional Σ-evaluation and $\boldsymbol{B}, \boldsymbol{C} \in \mathrm{wff}_\beta(\Sigma)$. Suppose $\mathcal{E}_\varphi(\boldsymbol{B}) \equiv \mathcal{E}_\varphi(\boldsymbol{C})$ for every assignment φ into $\mathcal{J}$. Then for all formulae $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, positions p, and assignments φ into $\mathcal{J}$, $\mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{B}]_p) \equiv \mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{C}]_p)$.*

PROOF. We show the assertion by an induction on the structure of $\boldsymbol{A}$. If p is the top position, we have

$$\mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{B}]_p) \equiv \mathcal{E}_\varphi(\boldsymbol{B}) \equiv \mathcal{E}_\varphi(\boldsymbol{C}) \equiv \mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{C}]_p).$$

In particular, if $\boldsymbol{A}$ is a constant or a variable, then p must be the top position and we are done. Otherwise, assume p is not the top position. If $\boldsymbol{A}$ is an application $\boldsymbol{FD}$, we have to consider two cases: $\boldsymbol{A}[\boldsymbol{B}]_p = \boldsymbol{F}[\boldsymbol{B}]_q\boldsymbol{D}$ and $\boldsymbol{A}[\boldsymbol{B}]_p = \boldsymbol{F}(\boldsymbol{D}[\boldsymbol{B}]_r)$ for some positions q and r. Since the second case is analogous we only show the first case. By the inductive hypothesis we have

$$\begin{aligned}\mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{B}]_p) &\equiv \mathcal{E}_\varphi(\boldsymbol{F}[\boldsymbol{B}]_q\boldsymbol{D}) \equiv \mathcal{E}_\varphi(\boldsymbol{F}[\boldsymbol{B}]_q)@\mathcal{E}_\varphi(\boldsymbol{D})\\ &\equiv \mathcal{E}_\varphi(\boldsymbol{F}[\boldsymbol{C}]_q)@\mathcal{E}_\varphi(\boldsymbol{D}) \equiv \mathcal{E}_\varphi(\boldsymbol{F}[\boldsymbol{C}]_q\boldsymbol{D}) \equiv \mathcal{E}_\varphi(\boldsymbol{A}[\boldsymbol{C}]_p).\end{aligned}$$

If $\boldsymbol{A}[\boldsymbol{B}]_p = \lambda X_\gamma.\boldsymbol{D}[\boldsymbol{B}]_q$, then we get the assertion from ξ-functionality. By the inductive hypothesis, we know $\mathcal{E}_\psi(\boldsymbol{D}[\boldsymbol{B}]_q) \equiv \mathcal{E}_\psi(\boldsymbol{D}[\boldsymbol{C}]_p)$ for every assignment ψ. In particular, for any assignment φ and $\mathsf{c} \in \mathcal{D}_\gamma$, we have $\mathcal{E}_{\varphi,[\mathsf{c}/X]}(\boldsymbol{D}[\boldsymbol{B}]_q) \equiv \mathcal{E}_{\varphi,[\mathsf{c}/X]}(\boldsymbol{D}[\boldsymbol{C}]_p)$.

By ξ-functionality, we have

$$\mathscr{E}_\varphi(\boldsymbol{A}[\boldsymbol{B}]_p) \equiv \mathscr{E}_\varphi(\lambda X\boldsymbol{.}\boldsymbol{D}[\boldsymbol{B}]_q) \equiv \mathscr{E}_\varphi(\lambda X\boldsymbol{.}\boldsymbol{D}[\boldsymbol{C}]_q) \equiv \mathscr{E}_\varphi(\boldsymbol{A}[\boldsymbol{C}]_p).$$

Thus we have completed all the cases and proven the assertion. ⊣

EXAMPLE 3.26 (Singleton evaluation). The singleton applicative structure (cf. Example 3.7) is a Σ-evaluation if for any assignment φ and formula $\boldsymbol{A}$ we take $\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathsf{a}$, where a is the (unique) member of $\mathscr{D}_\alpha$. Note that in this Σ-evaluation $\mathscr{E}(\lambda X\boldsymbol{.}X) \equiv \mathscr{E}_\varphi(\lambda X\boldsymbol{.}Y)$ for any assignment φ.

For a detailed discussion on the closure conditions needed for the domains for function types to be rich enough for evaluation functions to exist, we refer the reader to [2, 4].

Note that the applicative term structure $\mathrm{wff}(\Sigma)$ from Example 3.8 cannot be made into a Σ-evaluation by providing an evaluation function. To see this, suppose $\mathscr{E}$ is an evaluation function for $\mathrm{wff}(\Sigma)$ and $\boldsymbol{F} := \mathscr{E}(\lambda X_\alpha\boldsymbol{.}X) \in \mathrm{wff}_{\alpha\to\alpha}(\Sigma)$. Since $\mathscr{E}$ is assumed to be an evaluation function, we must have

$$\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathscr{E}_\varphi((\lambda X_\alpha\boldsymbol{.}X)\boldsymbol{A}) \equiv \boldsymbol{F}@\boldsymbol{A} \equiv \boldsymbol{F}\boldsymbol{A}$$

for every $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$. In particular, for any constant $a_\alpha \in \Sigma_\alpha$, we must have $\boldsymbol{F}a \equiv \mathscr{E}_\varphi(a) \equiv \mathscr{E}((\lambda X_\alpha\boldsymbol{.}X)a) \equiv \mathscr{E}(\lambda X_\alpha\boldsymbol{.}X)@\mathscr{E}(a) \equiv \boldsymbol{F}(\boldsymbol{F}a)$. But clearly $\boldsymbol{F}a \not\equiv \boldsymbol{F}(\boldsymbol{F}a)$ no matter what $\boldsymbol{F} \in \mathrm{wff}_{\alpha\to\alpha}(\Sigma)$ we choose. In particular, the "obvious" choice of $\mathscr{E}(\lambda X_\alpha\boldsymbol{.}X) \equiv (\lambda X_\alpha\boldsymbol{.}X)$ does not work. This example suggests that we need to consider β-convertible terms equal before we can obtain a term evaluation (cf. Definition 3.35).

DEFINITION 3.27 (Σ-evaluation congruences). A *congruence* on a Σ-evaluation $\mathscr{J} \equiv (\mathscr{D}, @, \mathscr{E})$ is a congruence on the underlying applicative structure $(\mathscr{D}, @)$. Given any two variable assignments φ and ψ into $(\mathscr{D}, @)$, we will use the notation $\varphi \sim \psi$ to indicate that $\varphi(X) \sim \psi(X)$ for every variable X.

A typed equivalence relation was defined to be a congruence if it respects application. In order to form a quotient of a Σ-evaluation, we must be able to define an evaluation function $\mathscr{E}^\sim$ on the quotient structure. But $\mathscr{E}^\sim$ interprets all terms, including λ-abstractions. It is not obvious that one can find a well-defined $\mathscr{E}^\sim$ that is really an evaluation function. In fact, the property one needs in order to show $\mathscr{E}^\sim$ will be a well-defined evaluation function is $\mathscr{E}_\varphi(\boldsymbol{A}) \sim \mathscr{E}_\psi(\boldsymbol{A})$ for all $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ and assignments φ and ψ with $\varphi \sim \psi$. One can show this by an easy induction on the term $\boldsymbol{A}$ if the congruence $\sim$ is functional. However, without the assumption that $\sim$ is functional, this direct proof will fail when $\boldsymbol{A}$ is a λ-abstraction. This is a general problem with trying to prove properties of evaluations since many objects in $\mathscr{D}_{\alpha\to\beta}$ may represent the same function from $\mathscr{D}_\alpha$ to $\mathscr{D}_\beta$. Fortunately, there is a way to use combinators to reduce such inductions to terms which only have very special λ-abstractions.

DEFINITION 3.28 (***SK***-combinatory formulae). For all types α, β, and γ, we define two families of closed formulae we call *combinators*:

$$\boldsymbol{K}_{\alpha\to\beta\to\alpha} := \lambda X_\alpha Y_\beta\boldsymbol{.}X$$
$$\boldsymbol{S}_{(\alpha\to\beta\to\gamma)\to(\alpha\to\beta)\to\alpha\to\gamma} := \lambda U_{\alpha\to\beta\to\gamma} V_{\alpha\to\beta} W_\alpha\boldsymbol{.}(UW(VW)).$$

We define the set of ***SK**-combinatory formulae* to be the least subset of the set $\bigcup_{\alpha\in\mathscr{T}} \mathrm{wff}_\alpha(\Sigma)$ containing every $\boldsymbol{K}$ and $\boldsymbol{S}$, every constant $c \in \Sigma$ and every variable, that is closed under application.

As shown in [3], every formula can be β-expanded to an $\boldsymbol{SK}$-combinatory formula.

LEMMA 3.29. *For every type α and $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, there is an $\boldsymbol{SK}$-combinatory formula $\boldsymbol{A}' \in \mathrm{wff}_\alpha(\Sigma)$ such that $\boldsymbol{A}'$ β-reduces to $\boldsymbol{A}$.*

PROOF. See Proposition 1 in [3]. The main difference to this setup is the signature, and this plays no role in the proof. ⊣

Now, we can show $\mathscr{E}_\varphi(\boldsymbol{A}) \sim \mathscr{E}_\psi(\boldsymbol{A})$ for $\boldsymbol{SK}$-combinatory $\boldsymbol{A}$ whenever $\varphi \sim \psi$.

LEMMA 3.30. *Let $\mathscr{J} \equiv (\mathscr{D}, @, \mathscr{E})$ be a Σ-evaluation, $\sim$ a congruence on $\mathscr{J}$, and φ and ψ assignments into $\mathscr{J}$ with $\varphi \sim \psi$. For every $\boldsymbol{SK}$-combinatory formula $\boldsymbol{A}$, we have $\mathscr{E}_\varphi(\boldsymbol{A}) \sim \mathscr{E}_\psi(\boldsymbol{A})$.*

PROOF. The proof is by induction on the $\boldsymbol{SK}$-combinatory formula $\boldsymbol{A}$. If $\boldsymbol{A}$ is a variable X, we have $\mathscr{E}_\varphi(X) \equiv \varphi(X) \sim \psi(X) \equiv \mathscr{E}_\psi(X)$. If $\boldsymbol{A}$ is closed (e.g., a constant in Σ or a combinator), then $\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathscr{E}_\psi(\boldsymbol{A})$, so certainly $\mathscr{E}_\varphi(\boldsymbol{A}) \sim \mathscr{E}_\psi(\boldsymbol{A})$. Finally, if $\boldsymbol{A}$ is an application of two $\boldsymbol{SK}$-combinatory formulae $\boldsymbol{F}$ and $\boldsymbol{B}$, then by the inductive hypothesis we have $\mathscr{E}_\varphi(\boldsymbol{F}) \sim \mathscr{E}_\psi(\boldsymbol{F})$ and $\mathscr{E}_\varphi(\boldsymbol{B}) \sim \mathscr{E}_\psi(\boldsymbol{B})$. Since $\sim$ respects application, $\mathscr{E}_\varphi(\boldsymbol{FB}) \equiv \mathscr{E}_\varphi(\boldsymbol{F})@\mathscr{E}_\varphi(\boldsymbol{B}) \sim \mathscr{E}_\psi(\boldsymbol{F})@\mathscr{E}_\psi(\boldsymbol{B}) \equiv \mathscr{E}_\psi(\boldsymbol{FB})$. ⊣

We can use this result to show the same property holds for all formulae.

LEMMA 3.31. *Let $\mathscr{J} \equiv (\mathscr{D}, @, \mathscr{E})$ be a Σ-evaluation, φ and ψ assignments into $\mathscr{J}$ with $\varphi \sim \psi$, and $\sim$ a congruence on $\mathscr{J}$. For every formula $\boldsymbol{A}$, we have $\mathscr{E}_\varphi(\boldsymbol{A}) \sim \mathscr{E}_\psi(\boldsymbol{A})$.*

PROOF. Let $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ for some type α. By Lemma 3.29 there is an $\boldsymbol{SK}$-combinatory formula $\boldsymbol{A}'$ that β-reduces to $\boldsymbol{A}$. By Remark 3.19 and Lemma 3.30, we have $\mathscr{E}_\varphi(\boldsymbol{A}) \equiv \mathscr{E}_\varphi(\boldsymbol{A}') \sim \mathscr{E}_\psi(\boldsymbol{A}') \equiv \mathscr{E}_\psi(\boldsymbol{A})$. ⊣

REMARK 3.32 (Correspondence with logical relations). Lemma 3.31 is essentially an instance of the "Basic Lemma" for logical relations (Lemma 8.2.5 in [44]). In fact, $\sim$ is functional, iff $\sim$ is a logical relation over the applicative structure. If $\sim$ is not functional, it still satisfies this "Basic Lemma" property, which makes it a pre-logical relation in the sense of [31].

DEFINITION 3.33 (Quotient Σ-evaluation). Let $\mathscr{J} \equiv (\mathscr{D}, @, \mathscr{E})$ be a Σ-evaluation, $\sim$ a congruence on $\mathscr{J}$ and let $(\mathscr{D}^\sim, @^\sim)$ be the quotient applicative structure of $(\mathscr{D}, @)$ with respect to $\sim$.

For each $\mathsf{A} \in \mathscr{D}_\alpha^\sim$, we choose a representative $\mathsf{A}^* \in \mathsf{A}$. So, $[\![\mathsf{A}^*]\!]_\sim \equiv \mathsf{A}$. Note that $[\![\mathsf{a}]\!]_\sim^* \sim \mathsf{a}$ for every $\mathsf{a} \in \mathscr{D}_\alpha$. For any assignment φ into $\mathscr{J}/_\sim$, let φ^* be the assignment into $\mathscr{J}$ given by $\varphi^*(X) := \varphi(X)^*$. Note that $\varphi \equiv \pi_\sim \circ \varphi^*$. So we can define $\mathscr{E}_\varphi^\sim$ as $\pi_\sim \circ \mathscr{E}_{\varphi^*}$, and call $\mathscr{J}/_\sim := (\mathscr{D}^\sim, @^\sim, \mathscr{E}^\sim)$ the *quotient Σ-evaluation* of $\mathscr{J}$ modulo $\sim$. (By Lemma 3.31, the definition of $\mathscr{E}^\sim$ does not depend on the choice of representatives.)

This definition is justified by the following theorem.

THEOREM 3.34 (Quotient Σ-evaluation theorem). *If $\mathscr{J}$ is a Σ-evaluation and $\sim$ is a congruence on $\mathscr{J}$, then $\mathscr{J}/_\sim$ is a Σ-evaluation.*

PROOF. We prove that $\mathscr{E}^\sim$ is an evaluation function by verifying the conditions in Definition 3.18. For any assignment φ into the quotient applicative structure, let

φ^* be the assignment with $\varphi \equiv \pi_\sim \circ \varphi^*$ as in Definition 3.33. First, we compute $\mathcal{E}^\sim_\varphi|_\mathcal{V} \equiv (\pi_\sim \circ \mathcal{E}_{\varphi^*})|_\mathcal{V} \equiv \pi_\sim \circ \mathcal{E}_{\varphi^*}|_\mathcal{V} \equiv \pi_\sim \circ \varphi^* \equiv \varphi$. Since $\pi_\sim$ is a homomorphism we have

$$\begin{aligned}
\mathcal{E}^\sim_\varphi(\boldsymbol{FA}) &\equiv \pi_\sim(\mathcal{E}_{\varphi^*}(\boldsymbol{FA}))\\
&\equiv \pi_\sim(\mathcal{E}_{\varphi^*}(\boldsymbol{F})@\mathcal{E}_{\varphi^*}(\boldsymbol{A}))\\
&\equiv \pi_\sim(\mathcal{E}_{\varphi^*}(\boldsymbol{F}))@^\sim\pi_\sim(\mathcal{E}_{\varphi^*}(\boldsymbol{A}))\\
&\equiv \mathcal{E}^\sim_\varphi(\boldsymbol{F})@^\sim\mathcal{E}^\sim_\varphi(\boldsymbol{A}).
\end{aligned}$$

If φ and ψ coincide on free($\boldsymbol{A}$), then $\mathcal{E}^\sim_\varphi(\boldsymbol{A}) \equiv [\![\mathcal{E}_{\varphi^*}(\boldsymbol{A})]\!]_\sim \equiv [\![\mathcal{E}_{\psi^*}(\boldsymbol{A})]\!]_\sim \equiv \mathcal{E}^\sim_\psi(\boldsymbol{A})$ since this entails that φ^* and ψ^* coincide on free($\boldsymbol{A}$) too (as we have chosen particular representatives for each equivalence class). Finally, $\mathcal{E}^\sim_\varphi(\boldsymbol{A}) \equiv [\![\mathcal{E}_{\varphi^*}(\boldsymbol{A})]\!]_\sim \equiv [\![\mathcal{E}_{\varphi^*}(\boldsymbol{A}{\downarrow_\beta})]\!]_\sim \equiv \mathcal{E}^\sim_\varphi(\boldsymbol{A}{\downarrow_\beta})$. ⊣

Definition 3.35 (Term evaluations for Σ). Let $\mathrm{cwff}(\Sigma){\downarrow_\beta}$ be the collection of closed well-formed formulae in β-normal form and $\boldsymbol{A}@^\beta\boldsymbol{B}$ be $(\boldsymbol{AB}){\downarrow_\beta}$. For the definition of an evaluation function let φ be an assignment into $\mathrm{cwff}(\Sigma){\downarrow_\beta}$. Note that $\sigma := \varphi|_{\mathrm{free}(\boldsymbol{A})}$ is a substitution, since free($\boldsymbol{A}$) is finite. Thus we can choose $\mathcal{E}^\beta_\varphi(\boldsymbol{A}) := \sigma(\boldsymbol{A}){\downarrow_\beta}$. We call $\mathcal{TE}(\Sigma)^\beta := (\mathrm{cwff}{\downarrow_\beta}, @^\beta, \mathcal{E}^\beta)$ the *β-term evaluation* for Σ.

Analogously, we can define $\mathcal{TE}(\Sigma)^{\beta\eta} := (\mathrm{cwff}{\downarrow_{\beta\eta}}, @^{\beta\eta}, \mathcal{E}^{\beta\eta})$ the *$\beta\eta$-term evaluation* for Σ.

The name *term evaluation* in the previous definition is justified by the following lemma.

Lemma 3.36. *$\mathcal{TE}(\Sigma)^\beta$ is a Σ-evaluation and $\mathcal{TE}(\Sigma)^{\beta\eta}$ is a functional Σ-evaluation.*

Proof. The fact that $(\mathrm{cwff}(\Sigma){\downarrow_\beta}, @^\beta)$ is an applicative structure is immediate: For each type α, $\mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta}$ is non-empty (by the assumption in Remark 3.16) and

$$@^\beta : \mathrm{cwff}_{\alpha\to\beta}(\Sigma){\downarrow_\beta} \times \mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta} \longrightarrow \mathrm{cwff}_\beta(\Sigma){\downarrow_\beta}.$$

We next check that $\mathcal{E}^\beta$ is an evaluation function.

(1) $\mathcal{E}^\beta_\varphi(X) \equiv \varphi|_{\mathrm{free}(X)}(X) \equiv \varphi(X)$.
(2) $\mathcal{E}^\beta_\varphi$ respects application since $\sigma(\boldsymbol{FA}){\downarrow_\beta} \equiv (\sigma(\boldsymbol{F}){\downarrow_\beta}\sigma(\boldsymbol{A}){\downarrow_\beta}){\downarrow_\beta}$ where $\sigma \equiv \varphi|_{\mathrm{free}(\boldsymbol{FA})}$.
(3) $\mathcal{E}^\beta_\varphi(\boldsymbol{A}) \equiv (\varphi|_{\mathrm{free}(\boldsymbol{A})}(\boldsymbol{A})){\downarrow_\beta} \equiv (\varphi'|_{\mathrm{free}(\boldsymbol{A})}(\boldsymbol{A})){\downarrow_\beta} \equiv \mathcal{E}^\beta_{\varphi'}(\boldsymbol{A})$ whenever φ and φ' coincide on free($\boldsymbol{A}$).
(4) $\mathcal{E}^\beta_\varphi(\boldsymbol{A}) \equiv \sigma(\boldsymbol{A}){\downarrow_\beta} \equiv \sigma(\boldsymbol{A}{\downarrow_\beta}){\downarrow_\beta} \equiv \mathcal{E}^\beta_\varphi(\boldsymbol{A}{\downarrow_\beta})$ where $\sigma \equiv \varphi|_{\mathrm{free}(\boldsymbol{A})}$.

A similar argument shows that $\mathcal{TE}(\Sigma)^{\beta\eta}$ is a Σ-evaluation. Also, one can show $\mathcal{TE}(\Sigma)^{\beta\eta}$ is functional using an argument similar to Lemma 3.14 since Σ is infinite at all types by Remark 3.16. (Alternatively, one can simply apply Lemma 3.14 and Theorem 3.13 to note that the applicative structure $\mathrm{cwff}(\Sigma)/_{\equiv_{\beta\eta}}$ is functional. The applicative structure $\mathrm{cwff}(\Sigma)/_{\equiv_{\beta\eta}}$ is isomorphic to the applicative structure

$(\mathrm{cwff}(\Sigma){\downarrow}_{\beta\eta}, @^{\beta\eta})$. One can easily show that functionality is preserved under isomorphism.) ⊣

REMARK 3.37. Note that $\mathcal{TE}(\Sigma)^\beta$ is not a functional Σ-evaluation since, for instance, for any constant $h_{\gamma\to\delta} \in \Sigma$

$$(\lambda X_\gamma . h_{\gamma\to\delta} X) @^\beta \boldsymbol{C}_\gamma \equiv h @^\beta \boldsymbol{C}$$

for all $\boldsymbol{C}$ in $\mathcal{TE}_\gamma(\Sigma)^\beta$ but $\lambda X . hX \not\equiv h$.

REMARK 3.38. One can show that an evaluation function $\mathcal{E}$ for an applicative structure $(\mathcal{D}, @)$ is uniquely determined by its values $\mathcal{E}(c)$ on the constants $c \in \Sigma$ and its values $\mathcal{E}(\boldsymbol{S})$ and $\mathcal{E}(\boldsymbol{K})$ on the combinators $\boldsymbol{S}$ and $\boldsymbol{K}$. When the applicative structure is functional, even the values of each $\mathcal{E}(\boldsymbol{S})$ and $\mathcal{E}(\boldsymbol{K})$ are determined, so that $\mathcal{E}$ is uniquely determined by its values $\mathcal{E}(c)$ for $c \in \Sigma$.

DEFINITION 3.39 (Homomorphism on Σ-evaluations). Let $\mathcal{J}^1 := (\mathcal{D}^1, @^1, \mathcal{E}^1)$ and $\mathcal{J}^2 := (\mathcal{D}^2, @^2, \mathcal{E}^2)$ be Σ-evaluations. A Σ-*homomorphism* is a typed function $\kappa\colon \mathcal{D}^1 \longrightarrow \mathcal{D}^2$ such that κ is a homomorphism from the applicative structure $(\mathcal{D}^1, @^1)$ to the applicative structure $(\mathcal{D}^2, @^2)$ and $\kappa\left(\mathcal{E}^1_\varphi(\boldsymbol{A})\right) \equiv \mathcal{E}^2_{\kappa\circ\varphi}(\boldsymbol{A})$ for every $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ and assignment φ for $\mathcal{J}^1$.

3.3. Σ-models. The semantic notions so far are independent of the set of base types. Now, we specialize these to obtain a notion of models by requiring specialized behavior on the type o of truth values. For this we use the notion of a Σ-valuation which gives a truth-value interpretation to the domain $\mathcal{D}_o$ of a Σ-evaluation consistent with the intuitive interpretations of the logical constants. Since models are semantic entities that are constructed primarily to make a statement about the truth or falsity of a formula, the requirement that there exists a Σ-valuation is perhaps the most general condition under which one wants to speak of a model. Thus we will define our most general notion of semantics as Σ-evaluations that have Σ-valuations.

DEFINITION 3.40. Fix two values $\mathtt{T} \not\equiv \mathtt{F}$. Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation and $\upsilon\colon \mathcal{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ be a (total) function. We define several properties that characterize logical operators with respect to υ in the table shown in Figure 2.

prop.	where	holds when			for all
$\mathfrak{L}_\neg(\mathsf{n})$	$\mathsf{n} \in \mathcal{D}_{o\to o}$	$\upsilon(\mathsf{n}@\mathsf{a}) \equiv \mathtt{T}$	iff	$\upsilon(\mathsf{a}) \equiv \mathtt{F}$	$\mathsf{a} \in \mathcal{D}_o$
$\mathfrak{L}_\vee(\mathsf{d})$	$\mathsf{d} \in \mathcal{D}_{o\to o\to o}$	$\upsilon(\mathsf{d}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$	iff	$\upsilon(\mathsf{a}) \equiv \mathtt{T}$ or $\upsilon(\mathsf{b}) \equiv \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\wedge(\mathsf{c})$	$\mathsf{c} \in \mathcal{D}_{o\to o\to o}$	$\upsilon(\mathsf{c}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$	iff	$\upsilon(\mathsf{a}) \equiv \mathtt{T}$ and $\upsilon(\mathsf{b}) \equiv \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\Rightarrow(\mathsf{i})$	$\mathsf{i} \in \mathcal{D}_{o\to o\to o}$	$\upsilon(\mathsf{i}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$	iff	$\upsilon(\mathsf{a}) \equiv \mathtt{F}$ or $\upsilon(\mathsf{b}) \equiv \mathtt{T}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}_\Leftrightarrow(\mathsf{e})$	$\mathsf{e} \in \mathcal{D}_{o\to o\to o}$	$\upsilon(\mathsf{e}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$	iff	$\upsilon(\mathsf{a}) \equiv \upsilon(\mathsf{b})$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$
$\mathfrak{L}^\alpha_\forall(\pi)$	$\pi \in \mathcal{D}_{(\alpha\to o)\to o}$	$\upsilon(\pi@\mathsf{f}) \equiv \mathtt{T}$	iff	$\forall \mathsf{a} \in \mathcal{D}_\alpha\ \upsilon(\mathsf{f}@\mathsf{a}) \equiv \mathtt{T}$	$\mathsf{f} \in \mathcal{D}_{\alpha\to o}$
$\mathfrak{L}^\alpha_\exists(\sigma)$	$\sigma \in \mathcal{D}_{(\alpha\to o)\to o}$	$\upsilon(\sigma@\mathsf{f}) \equiv \mathtt{T}$	iff	$\exists \mathsf{a} \in \mathcal{D}_\alpha\ \upsilon(\mathsf{f}@\mathsf{a}) \equiv \mathtt{T}$	$\mathsf{f} \in \mathcal{D}_{\alpha\to o}$
$\mathfrak{L}^\alpha_=(\mathsf{q})$	$\mathsf{q} \in \mathcal{D}_{\alpha\to\alpha\to o}$	$\upsilon(\mathsf{q}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$	iff	$\mathsf{a} \equiv \mathsf{b}$	$\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$

FIGURE 2. Logical properties in Σ-models.

DEFINITION 3.41 (Σ-model). Let $\mathcal{J} := (\mathcal{D}, @, \mathcal{E})$ be a Σ-evaluation. A function $\upsilon\colon \mathcal{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ is called a Σ-*valuation* for $\mathcal{J}$ if $\mathfrak{L}_\neg(\mathcal{E}(\neg))$ and $\mathfrak{L}_\vee(\mathcal{E}(\vee))$ hold, and for every type α $\mathfrak{L}^{\alpha}_{\forall}(\mathscr{E}(\Pi^{\alpha}))$ holds. In this case, $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ is called a Σ-*model*.

For the case of (the optional) primitive equality, i.e., when $=^{\alpha} \in \Sigma_{\alpha\to\alpha\to o}$ for all types α, we say $\mathscr{M}$ is a Σ-*model with primitive equality* if $\mathfrak{L}^{\alpha}_{=}(\mathscr{E}(=^{\alpha}))$ holds for every type α.

We say that φ is an assignment into $\mathscr{M}$ if it is an assignment into the underlying applicative structure $(\mathscr{D}, @)$. Furthermore, φ *satisfies* a formula $\boldsymbol{A} \in \mathrm{wff}_o(\Sigma)$ in $\mathscr{M}$ (we write $\mathscr{M} \models_{\varphi} \boldsymbol{A}$) if $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{A})) \equiv \mathtt{T}$. We say that $\boldsymbol{A}$ is *valid* in $\mathscr{M}$ (and write $\mathscr{M} \models \boldsymbol{A}$) if $\mathscr{M} \models_{\varphi} \boldsymbol{A}$ for all assignments φ. When $\boldsymbol{A} \in \mathrm{cwff}_o(\Sigma)$, we drop the reference to the assignment and use the notation $\mathscr{M} \models \boldsymbol{A}$. Finally, we say that $\mathscr{M}$ is a Σ-*model* for a set $\Phi \subseteq \mathrm{cwff}_o(\Sigma)$ (we write $\mathscr{M} \models \Phi$) if $\mathscr{M} \models \boldsymbol{A}$ for all $\boldsymbol{A} \in \Phi$.

A Σ-model $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ is called *functional* [*full*, *standard*] if the applicative structure $(\mathscr{D}, @)$ is *functional* [*full*, *standard*]. Similarly, $\mathscr{M}$ is called η-*functional* [ξ-*functional*] if the evaluation $(\mathscr{D}, @, \mathscr{E})$ is η-*functional* [ξ-*functional*]. We say $\mathscr{M}$ is a Σ-model over a frame if $(\mathscr{D}, @)$ is a frame.

REMARK 3.42 (Adding primitive equality). In the definition of Σ-model above, the addition of property $\mathfrak{L}^{\alpha}_{=}(\mathscr{E}(=^{\alpha}))$ addressing the case of primitive equality above has a purely practical motivation: calculi with a primitive treatment of equality, see for instance [10, 11], may provide a more effective approach to equational reasoning in higher-order logic than the exclusive use of Leibniz equality. Therefore we enrich our theory to automatically also address the situation where (always built-in) Leibniz equality and (optional) primitive equality are simultaneously present in the language. The generalization to primitive equality is less trivial than the generalization to other (optional) primitive logical connectives such as $\wedge$ or $\Rightarrow$. This is the main reason why we built primitive equality directly into our theory while we omit other logical primitives (cf. also Remarks 3.47 and 6.9).

LEMMA 3.43 (Truth and falsity in Σ-models). *Let $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ be a Σ-model and φ an assignment. Let $\boldsymbol{T}_o := \forall P_o . P \vee \neg P$ and $\boldsymbol{F}_o := \neg \boldsymbol{T}_o$. Then $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{T}_o)) \equiv \mathtt{T}$ and $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{F}_o)) \equiv \mathtt{F}$.*

PROOF. Let P be a variable of type o. We have $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{T}_o)) \equiv \mathtt{T}$, iff $\upsilon(\mathscr{E}_{\varphi}(P \vee \neg P)) \equiv \mathtt{T}$ for every assignment φ. The properties of υ show that this statement is equivalent to $\upsilon(\varphi(P)) \equiv \mathtt{T}$ or $\upsilon(\varphi(P)) \equiv \mathtt{F}$, which is always true since υ maps into $\{\mathtt{T}, \mathtt{F}\}$. Note further that $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{F}_o)) \equiv \mathtt{F}$ since $\upsilon(\mathscr{E}_{\varphi}(\boldsymbol{T}_o)) \equiv \mathtt{T}$. ⊣

REMARK 3.44. Let $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ be a Σ-model. By Lemma 3.43, $\mathscr{D}_o$ must have at least the two elements $\mathscr{E}_{\varphi}(\boldsymbol{T}_o)$ and $\mathscr{E}_{\varphi}(\boldsymbol{F}_o)$, and υ must be surjective.

REMARK 3.45. In contrast to the case of Henkin models, Definition 3.41 only constrains the functional behavior of the values of the logical constants with respect to υ. This does not fully specify these values since

- $\mathscr{M}$ need not be functional,
- and there can be more than two truth values.

We will now introduce semantical properties called $\mathfrak{q}$, η, $\mathfrak{f}$, and $\mathfrak{b}$, which we will use to characterize different classes of Σ-models.

DEFINITION 3.46 (Properties $\mathfrak{q}$, η, ξ, $\mathfrak{f}$ and $\mathfrak{b}$). Given a Σ-model $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$, we say that $\mathscr{M}$ has *property*

$\mathfrak{q}$: iff for all $\alpha \in \mathscr{T}$ there is some $\mathsf{q}^\alpha \in \mathscr{D}_{\alpha\to\alpha\to o}$ such that $\mathfrak{L}^\alpha_=(\mathsf{q}^\alpha)$ holds.
η: iff $\mathscr{M}$ is η-functional.
ξ: iff $\mathscr{M}$ is ξ-functional.
$\mathfrak{f}$: iff $\mathscr{M}$ is functional. (This is generally associated with functional extensionality.)
$\mathfrak{b}$: iff $\mathscr{D}_o$ has at most two elements. By Lemma 3.44 we can assume without loss of generality that $\mathscr{D}_o \equiv \{\mathtt{T}, \mathtt{F}\}$, υ is the identity function, $\mathscr{E}_\varphi(\boldsymbol{T}_o) \equiv \mathtt{T}$ and $\mathscr{E}_\varphi(\boldsymbol{F}_o) \equiv \mathtt{F}$. (This is generally associated with Boolean extensionality.)

Remark 3.47 (Choice of logical constants). The work presented in this article is based on the choice of the primitive logical constants $\neg$, $\vee$, and Π^α. We have also introduced shorthand for formulas constructed using $\wedge$, $\Rightarrow$, $\Leftrightarrow$, and existential quantification. One can (easily; cf. Lemma 3.48) verify that in any Σ-model $\mathscr{M} \equiv (\mathscr{D}, @, \mathscr{E}, \upsilon)$, each of the properties $\mathfrak{L}_\wedge(\mathscr{E}(\lambda X_o Y_o \boldsymbol{.} X \wedge Y))$, $\mathfrak{L}_\Rightarrow(\mathscr{E}(\lambda X_o Y_o \boldsymbol{.} X \Rightarrow Y))$, $\mathfrak{L}_\Leftrightarrow(\mathscr{E}(\lambda X_o Y_o \boldsymbol{.} X \Leftrightarrow Y))$ and $\mathfrak{L}^\alpha_\exists(\mathscr{E}(\lambda P_{\alpha\to o} \boldsymbol{.} \exists X_\alpha \boldsymbol{.} PX))$ (for each type α) hold with respect to υ. In this sense, our choice of logical constants and shorthand for other logical constants is sufficient. However, Leibniz equality $\mathbf{Q}^\alpha$ will only satisfy $\mathfrak{L}^\alpha_=(\mathscr{E}(\mathbf{Q}^\alpha))$ for each type α iff the model satisfies property $\mathfrak{q}$ (cf. Remark 3.52 and Theorem 3.63).

On the other hand, in the absence of extensionality, one can gain some (limited) expressive power by including extra logical constants such as $\wedge$ in the signature. This is the case since there may be several objects in $\mathsf{c} \in \mathscr{D}_{o\to o\to o}$ such that $\mathfrak{L}_\wedge(\mathsf{c})$ holds. So, one could have a Σ-model $\mathscr{M} \equiv (\mathscr{D}, @, \mathscr{E}, \upsilon)$ (where $\wedge$ is also in Σ) such that $\mathfrak{L}_\wedge(\mathscr{E}(\wedge))$ holds, but $\mathscr{E}(\wedge) \not\equiv \mathscr{E}(\lambda X_o Y_o \boldsymbol{.} \neg(\neg X \vee \neg Y))$. We will not investigate this possibility here.

Our choice of logical constants differs from Andrews' choice [6] who considers primitive equality as the only logical primitive from which all other logical operators are defined using the definitions in Figure 3. For the sake of clarity, we write $\mathbf{q}^\alpha$ for $=^\alpha$ when $=^\alpha$ is not being written in infix notation. For Henkin models, the definitions in Figure 3 are appropriate. However, without extensionality, the situation is quite different. Suppose $\mathscr{J} \equiv (\mathscr{D}, @, \mathscr{E})$ is a Σ-evaluation where $=^\alpha \in \Sigma$ for every type α. Let $\upsilon : \mathscr{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ be a function such that $\mathfrak{L}^\alpha_=(\mathscr{E}(=^\alpha))$ holds for each type α. The fact that $\upsilon(\mathscr{E}(\boldsymbol{T}_o)) \equiv \mathtt{T}$ follows directly from $\mathfrak{L}^{o\to o\to o}_=(\mathscr{E}(=^{o\to o\to o}))$ and reflexivity of (meta-level) equality. Unfortunately, this is the last definition which is clearly appropriate without further assumptions. So long as $\mathscr{D}_o$ has more than one element, one can show $\upsilon(\mathscr{E}(\boldsymbol{F}_o)) \equiv \mathtt{F}$. So, let us explicitly assume $\mathscr{D}_o$

$\boldsymbol{T}_o$	$:=$	$\mathbf{q}^o =^{o\to o\to o} \mathbf{q}^o$
$\boldsymbol{F}_o$	$:=$	$(\lambda X_o \boldsymbol{.} \boldsymbol{T}_o) =^{o\to o} (\lambda X_o \boldsymbol{.} X)$
$\neg_{o\to o}$	$:=$	$\mathbf{q}^o \boldsymbol{F}_o$
Π^α	$:=$	$\mathbf{q}^{\alpha\to o}(\lambda X_\alpha \boldsymbol{.} \boldsymbol{T}_o)$
$\wedge_{o\to o\to o}$	$:=$	$\lambda X_o Y_o \boldsymbol{.} (\lambda G_{o\to o\to o} \boldsymbol{.} G\, \boldsymbol{T}_o \boldsymbol{T}_o) =^{(o\to o\to o)\to o} (\lambda G_{o\to o\to o} \boldsymbol{.} GXY)$
$\Rightarrow_{o\to o\to o}$	$:=$	$\lambda X_o Y_o \boldsymbol{.} (X =^o (X \wedge Y))$
$\vee_{o\to o\to o}$	$:=$	$\lambda X_o Y_o \boldsymbol{.} \neg(\neg X \wedge \neg Y)$
Σ^α	$:=$	$\lambda P_{\alpha\to o} \boldsymbol{.} (\neg \Pi^\alpha \lambda X_\alpha \boldsymbol{.} \neg(PX))$

Figure 3. A definition of logical constants from equality in Henkin models.

has more than one element, which is anyway met by Σ-models (cf. Remark 3.44). Next, we investigate whether $\mathfrak{L}_\neg(\mathscr{E}(\neg))$ holds. Let $\mathsf{a} \in \mathscr{D}_o$ be given. By $\mathfrak{L}^o_=(\mathscr{E}(=^o))$, we know $\upsilon(\mathscr{E}(=^o)@\mathscr{E}(\boldsymbol{F}_o)@\mathsf{a}) \equiv \mathtt{T}$ is equivalent to $\mathscr{E}(\boldsymbol{F}_o) \equiv \mathsf{a}$. So, if $\upsilon(\mathscr{E}(=^o)@\mathscr{E}(\boldsymbol{F}_o)@\mathsf{a}) \equiv \mathtt{T}$, then $\upsilon(\mathsf{a}) \equiv \upsilon(\mathscr{E}(\boldsymbol{F}_o)) \equiv \mathtt{F}$. For the converse, suppose $\upsilon(\mathsf{a}) \equiv \mathtt{F}$. This, in general, does not imply $\mathscr{E}(\boldsymbol{F}_o) \equiv \mathsf{a}$. However, if we assume a is the *unique* member of $\mathscr{D}_o$ such that $\upsilon(\mathsf{a}) \equiv \mathtt{F}$, then we can conclude $\mathscr{E}(\boldsymbol{F}_o) \equiv \mathsf{a}$. In particular, if $\mathscr{D}_o$ has only two elements, then υ must be injective and we can conclude $\mathscr{E}(\boldsymbol{F}_o) \equiv \mathsf{a}$. So, Boolean extensionality is required to ensure that $\mathfrak{L}_\neg(\mathscr{E}(\neg))$ holds for this definition of $\neg$.

We now investigate whether $\mathfrak{L}^\alpha_\forall(\mathscr{E}(\Pi^\alpha))$ holds for Π^α defined as in Figure 3. Let $\mathsf{f} \in \mathscr{D}_{\alpha\to o}$ be given. Suppose $\upsilon(\mathscr{E}(=^{\alpha\to o})@\mathscr{E}(\lambda X_\alpha.\boldsymbol{T}_o)@\mathsf{f}) \equiv \mathtt{T}$. Then, by $\mathfrak{L}^{\alpha\to o}_=(\mathscr{E}(=^{\alpha\to o}))$, we know $\mathscr{E}(\lambda X_\alpha.\boldsymbol{T}_o) \equiv \mathsf{f}$. This does guarantee $\mathscr{E}(\boldsymbol{T}_o) \equiv \mathsf{f}@\mathsf{a}$ and hence $\upsilon(\mathsf{f}@\mathsf{a}) \equiv \mathtt{T}$ for every $\mathsf{a} \in \mathscr{D}_\alpha$. However, showing the converse requires that $\mathscr{M}$ is functional (i.e., strong functional extensionality is given). Suppose $\upsilon(\mathscr{E}(=^\alpha)@\mathscr{E}(\lambda X_\alpha.\boldsymbol{T}_o)@\mathsf{f}) \equiv \mathtt{F}$. We can conclude $\mathscr{E}(\lambda X_\alpha.\boldsymbol{T}_o) \not\equiv \mathsf{f}$, but this is of little value. If $\mathscr{J}$ is not functional, then these may be different representatives in $\mathscr{D}_{\alpha\to o}$ of the same function. If $\mathscr{J}$ is functional, there must be some $\mathsf{a} \in \mathscr{D}_\alpha$ such that $\mathscr{E}(\boldsymbol{T}_o) \not\equiv \mathsf{f}@\mathsf{a}$. However, this still does not imply $\upsilon(\mathsf{f}@\mathsf{a}) \equiv \mathtt{F}$. If $\mathscr{D}_o$ has only two elements, then the facts that $\mathscr{E}(\boldsymbol{T}_o) \not\equiv \mathsf{f}@\mathsf{a}$ and $\mathscr{E}(\boldsymbol{T}_o) \not\equiv \mathscr{E}(\boldsymbol{F}_o)$ imply $\mathscr{E}(\boldsymbol{F}_o) \equiv \mathsf{f}@\mathsf{a}$, hence $\upsilon(\mathsf{f}@\mathsf{a}) \equiv \mathtt{F}$.

Similar observations apply to the other definitions in Figure 3. These definitions do show that at least $\boldsymbol{T}_o$ and $\boldsymbol{F}_o$ are definable from primitive equality (so long as $\mathscr{D}_o$ has at least two elements). Furthermore, if $\mathscr{D}_o$ has exactly two elements $\neg$ is definable from primitive equality. We conjecture that this is as much as one can define in terms of primitive equality without extensionality assumptions. That is, we conjecture that without assuming $\mathscr{D}_o$ has two elements, there may be no object $\mathsf{n} \in \mathscr{D}_{o\to o}$ such that $\mathfrak{L}_\neg(\mathsf{n})$ holds. Furthermore, we conjecture that without assuming functionality and that $\mathscr{D}_o$ has two elements, there may be no object $\mathsf{d} \in \mathscr{D}_{o\to o\to o}$ such that $\mathfrak{L}_\vee(\mathsf{d})$ holds, and there may be no object $\pi \in \mathscr{D}_{(\alpha\to o)\to o}$ such that $\mathfrak{L}^\alpha_\forall(\pi)$ holds.

The next lemma formally verifies that $\mathfrak{L}_\Leftrightarrow(\mathscr{E}(\lambda X_o Y_o.X \Leftrightarrow Y))$ holds with respect to the valuation of a Σ-model, as indicated in the remark above.

Lemma 3.48 (Equivalence). *Let $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ be a Σ-model, φ an assignment into $\mathscr{M}$, and $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_o(\Sigma)$. $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A} \Leftrightarrow \boldsymbol{B})) \equiv \mathtt{T}$ iff $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B}))$.*

Proof. Suppose $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A} \Leftrightarrow \boldsymbol{B})) \equiv \mathtt{T}$. This implies $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{A} \vee \boldsymbol{B})) \equiv \mathtt{T}$ and $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{B}\vee\boldsymbol{A})) \equiv \mathtt{T}$. If $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \mathtt{T}$, then $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{A}\vee\boldsymbol{B})) \equiv \mathtt{T}$ implies $\upsilon(\mathscr{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{T}$, so $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \mathtt{T} \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B}))$. If $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \mathtt{F}$, then $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{B} \vee \boldsymbol{A})) \equiv \mathtt{T}$ implies $\upsilon(\mathscr{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{F}$, so $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \mathtt{F} \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B}))$. Since these are the only two possible values for $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A}))$, we have $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B}))$.

Suppose $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B}))$. Either $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{T}$ or $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathscr{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{F}$. An easy consideration of both cases verifies $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{A} \vee \boldsymbol{B})) \equiv \mathtt{T}$ and $\upsilon(\mathscr{E}_\varphi(\neg\boldsymbol{B} \vee \boldsymbol{A})) \equiv \mathtt{T}$. Hence, $\upsilon(\mathscr{E}_\varphi(\boldsymbol{A} \Leftrightarrow \boldsymbol{B})) \equiv \mathtt{T}$. ⊣

We next define classes of Σ-models in which certain properties hold. These classes are denoted by $\mathfrak{M}_*$ where $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. The subscript β is always included to emphasize that β-equal terms are interpreted to be identical elements in all models (cf. Remark 3.19). The subscripts η, ξ, $\mathfrak{f}$ and $\mathfrak{b}$ indicate when the corresponding properties must hold (cf. Definition 3.46). Note that we are not including property $\mathfrak{q}$ as an explicit subscript. The only Σ-models we need to consider which do not satisfy property $\mathfrak{q}$ are term models. It will turn out (cf. Theorem 3.62) that we can obtain a model satisfying property $\mathfrak{q}$ from a model that does not by taking a quotient. However, this may not preserve properties ξ or $\mathfrak{f}$. Consequently, we omit $\mathfrak{q}$ as a subscript and define the sets $\mathfrak{M}_*$ (for $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$) so that every model in $\mathfrak{M}_*$ satisfies property $\mathfrak{q}$. (This choice will be discussed further in Remark 3.52.)

DEFINITION 3.49 (Higher-order model classes). We will denote the class of Σ-models that satisfy property $\mathfrak{q}$ by $\mathfrak{M}_\beta$, and we will use subclasses of $\mathfrak{M}_\beta$ depending on the validity of the properties η, ξ, $\mathfrak{f}$, and $\mathfrak{b}$. We obtain the specialized classes of Σ-models $\mathfrak{M}_{\beta\eta}$, $\mathfrak{M}_{\beta\xi}$, $\mathfrak{M}_{\beta\mathfrak{f}}$, $\mathfrak{M}_{\beta\mathfrak{b}}$, $\mathfrak{M}_{\beta\eta\mathfrak{b}}$, $\mathfrak{M}_{\beta\xi\mathfrak{b}}$, and $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ by requiring that the properties specified in the index are valid.

If primitive equality is in the signature, i.e., if $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$, then we require the models to be Σ-models with primitive equality. Note that in this case property $\mathfrak{q}$ is automatically ensured.

We can group these eight classes in two dimensions as in Figure 4 based on the "amount of extensionality" required.

	functional				
Boolean		none	weak (η)	weak (ξ)	strong ($\mathfrak{f}$)
	none	$\mathfrak{M}_\beta$	$\mathfrak{M}_{\beta\eta}$	$\mathfrak{M}_{\beta\xi}$	$\mathfrak{M}_{\beta\mathfrak{f}}$
	$\mathfrak{b}$	$\mathfrak{M}_{\beta\mathfrak{b}}$	$\mathfrak{M}_{\beta\eta\mathfrak{b}}$	$\mathfrak{M}_{\beta\xi\mathfrak{b}}$	$\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$

FIGURE 4. Extensional model classes.

DEFINITION 3.50 (Σ-Henkin models). A Σ-*Henkin model* is a model $\mathcal{M}$ over a frame with $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$. We denote the class of all Σ-Henkin models by $\mathfrak{H}$. (Such models are called *general models* in [2] and [6]. We avoid this terminology here since we consider models which are more general than these.)

DEFINITION 3.51 (Σ-standard models). A Σ-*standard model* is a Σ-Henkin model that is also full (i.e., a model $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ over a standard frame). The class of all Σ-standard models is denoted by $\mathfrak{ST}$.

REMARK 3.52 (Property $\mathfrak{q}$). The purpose of property $\mathfrak{q}$ is to ensure that for all types α there is an object q^α in $\mathcal{D}_{\alpha\to\alpha\to o}$ representing meta equality for the domain $\mathcal{D}_\alpha$. This ensures the existence of objects representing unit sets $\{\mathsf{a}\}$ for each $\mathsf{a} \in \mathcal{D}_\alpha$ in the domains $\mathcal{D}_{\alpha\to o}$, which in turn makes Leibniz equality the intended equality relation. This is because membership in these unit sets can be used as an appropriately strong criterion to distinguish between different elements of $\mathcal{D}_\alpha$. This aspect is discussed in detail by Peter Andrews in [2]. He notes that Leon Henkin unintentionally introduced in [26] a class of models which need not satisfy property $\mathfrak{q}$ instead of the class of Henkin models in the sense above. As Andrews shows, a consequence is that such a model may fail to satisfy the principle of strong functional extensionality (cf. Definition 4.5) given by the formula

$$\forall F_{\iota\to\iota}. \forall G_{\iota\to\iota}. (\forall X_\iota. FX \doteq^\iota GX) \Rightarrow F \doteq^{\iota\to\iota} G$$

even though the model (as a model over a frame) is functional. Andrews fixed this problem by introducing property $\mathfrak{q}$. Here, we have followed this by requiring property $\mathfrak{q}$ in all our model classes $\mathfrak{M}_*$.

Now let us extend the notion of a quotient evaluation to Σ-models.

DEFINITION 3.53 (Σ-model congruences). A *congruence* on a Σ-model $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ is a congruence on the underlying Σ-evaluation $(\mathcal{D}, @, \mathcal{E})$ such that $\upsilon(\mathsf{a}) \equiv \upsilon(\mathsf{b})$ for all $\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$ with $\mathsf{a} \sim \mathsf{b}$.

DEFINITION 3.54 (Quotient Σ-model). Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ be a Σ-model, $\sim$ be a congruence on $\mathcal{M}$, and $(\mathcal{D}^\sim, @^\sim, \mathcal{E}^\sim)$ be the quotient Σ-evaluation of $(\mathcal{D}, @, \mathcal{E})$ with respect to $\sim$ (cf. Definition 3.33). Using the notation for representatives $\mathsf{A}^* \in \mathsf{A}$ for $\mathsf{A} \in \mathcal{D}_\alpha^\sim$ as in Definition 3.33, we define $\upsilon^\sim : \mathcal{D}_o^\sim \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ by $\upsilon^\sim(\mathsf{A}) := \upsilon(\mathsf{A}^*)$ for every $\mathsf{A} \in \mathcal{D}_o^\sim$. (Since $\upsilon(\mathsf{a}) \equiv \upsilon(\mathsf{b})$ whenever $\mathsf{a} \sim \mathsf{b}$ in $\mathcal{D}_o$, this definition of $\upsilon^\sim$ does not depend on the choice of representatives and $\upsilon^\sim([\![\mathsf{a}]\!]_\sim) \equiv \upsilon(\mathsf{a})$ for every $\mathsf{a} \in \mathcal{D}_o$.) We call $\mathcal{M}/_\sim := (\mathcal{D}^\sim, @^\sim, \mathcal{E}^\sim, \upsilon^\sim)$ the *quotient* Σ-model of $\mathcal{M}$ with respect to $\sim$.

THEOREM 3.55 (Quotient Σ-model theorem). *Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ be a Σ-model and $\sim$ be a congruence on $\mathcal{M}$. The quotient $\mathcal{M}/_\sim$ is a Σ-model.*

Furthermore, if for every type α, $=^\alpha \in \Sigma_\alpha$ and we have $\upsilon(\mathcal{E}(=^\alpha)@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$ iff $\mathsf{a} \sim \mathsf{b}$ for every $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$, then $\mathcal{M}/_\sim$ is a Σ-model with primitive equality.

PROOF. We check the conditions of Definition 3.41, again using the A^* notation for representatives. To check condition $\mathfrak{L}_\neg(\mathcal{E}^\sim(\neg))$ for $\upsilon^\sim$, for all $\mathsf{A} \in \mathcal{D}_o^\sim$ we need to show that $\upsilon^\sim(\mathcal{E}^\sim(\neg)@^\sim\mathsf{A}) \equiv \mathtt{T}$ iff $\upsilon^\sim(\mathsf{A}) \equiv \mathtt{F}$. Let $\mathsf{A} \in \mathcal{D}_o^\sim$ be given. Since $\mathcal{M}$ is a Σ-model we have $\upsilon(\mathcal{E}(\neg)@\mathsf{A}^*) \equiv \mathtt{T}$ iff $\upsilon(\mathsf{A}^*) \equiv \mathtt{F}$. Since $[\![\mathsf{A}^*]\!]_\sim \equiv \mathsf{A}$ and $[\![\mathcal{E}(\neg)@\mathsf{A}^*]\!]_\sim \equiv \mathcal{E}^\sim(\neg)@^\sim\mathsf{A}$, we have $\upsilon^\sim(\mathcal{E}^\sim(\neg)@^\sim\mathsf{A}) \equiv \mathtt{T}$ iff $\upsilon^\sim(\mathsf{A}) \equiv \mathtt{F}$. Checking condition $\mathfrak{L}_\vee(\mathcal{E}^\sim(\vee))$ for $\upsilon^\sim$ is analogous.

To check condition $\mathfrak{L}_\forall^\alpha(\mathcal{E}^\sim(\Pi^\alpha))$ for $\upsilon^\sim$, suppose we have $\mathsf{G} \in \mathcal{D}_{\alpha\to o}^\sim$. For every $\mathsf{A} \in \mathcal{D}_\alpha^\sim$, $\upsilon^\sim(\mathsf{G}@^\sim\mathsf{A}) \equiv \upsilon(\mathsf{G}^*@\mathsf{A}^*)$. So, if $\upsilon^\sim(\mathsf{G}@^\sim\mathsf{A}) \equiv \mathtt{T}$ for every $\mathsf{A} \in \mathcal{D}_\alpha^\sim$, then $\upsilon(\mathsf{G}^*@\mathsf{a}) \equiv \upsilon(\mathsf{G}^*@[\![\mathsf{a}]\!]_\sim^*) \equiv \mathtt{T}$ for every $\mathsf{a} \in \mathcal{D}_\alpha$, and we conclude $\upsilon(\mathcal{E}(\Pi^\alpha)@\mathsf{G}^*) \equiv \mathtt{T}$. Hence, $\upsilon^\sim(\mathcal{E}^\sim(\Pi^\alpha)@^\sim\mathsf{G}) \equiv \mathtt{T}$. Conversely, suppose $\upsilon^\sim(\mathcal{E}^\sim(\Pi^\alpha)@\mathsf{G}) \equiv \mathtt{T}$. Then $\upsilon(\mathcal{E}(\Pi^\alpha)@\mathsf{G}^*) \equiv \mathtt{T}$ and hence $\upsilon^\sim(\mathsf{G}@\mathsf{A}) \equiv \upsilon(\mathsf{G}^*@\mathsf{A}^*) \equiv \mathtt{T}$ for every $\mathsf{A} \in \mathcal{D}_\alpha^\sim$.

Suppose primitive equality is in the signature and $\upsilon(\mathcal{E}(=^\alpha)@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$ iff $\mathsf{a} \sim \mathsf{b}$ for every $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$. To verify $\mathfrak{L}_=^\alpha(\mathcal{E}^\sim(=^\alpha))$ holds for $\upsilon^\sim$, we simply note that $\upsilon^\sim(\mathcal{E}^\sim(=^\alpha)@^\sim\mathsf{A}@^\sim\mathsf{B}) \equiv \mathtt{T}$, iff $\upsilon(\mathcal{E}(=^\alpha)@\mathsf{A}^*@\mathsf{B}^*) \equiv \mathtt{T}$, iff $\mathsf{A}^* \sim \mathsf{B}^*$, iff $\mathsf{A} \equiv \mathsf{B}$. $\dashv$

We can define properties of a congruence analogous to those defined for models in Definition 3.46.

DEFINITION 3.56 (Properties η, ξ, $\mathfrak{f}$ and $\mathfrak{b}$ for congruences). Given a Σ-model $\mathcal{M} := (\mathcal{D}, @, \mathcal{E}, \upsilon)$ and a congruence $\sim$ on $\mathcal{M}$, we say $\sim$ has *property*

- η: iff $\mathcal{E}_\varphi(\boldsymbol{A}) \sim \mathcal{E}_\varphi(\boldsymbol{A}\!\downarrow_{\beta\eta})$ for any type α, $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, and assignment φ.
- ξ: iff for all $\alpha, \beta \in \mathcal{T}$, $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, assignment φ, and variables X_α, $\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{M}_\beta) \sim \mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{N}_\beta)$ whenever $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \sim \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$.
- $\mathfrak{f}$: iff $\sim$ is functional.
- $\mathfrak{b}$: iff $\mathcal{D}_o$ has at most two equivalence classes with respect to $\sim$. (By Remark 3.44 there are always at least two.)

REMARK 3.57. It follows trivially from reflexivity of congruences that if a model satisfies property η, then any congruence on the model satisfies property η. Similarly, if a model has only two elements in $\mathcal{D}_o$, then $\mathcal{D}_o$ can have at most two equivalence classes with respect to any congruence $\sim$. So, if a model satisfies property $\mathfrak{b}$, then any congruence on the model satisfies property $\mathfrak{b}$. This is not true for properties ξ or $\mathfrak{f}$. For an example, we refer to the functional model (satisfying property $\mathfrak{f}$, hence property ξ) constructed by Andrews in [2]. Using the results we prove below, one can show Leibniz equality must induce a congruence failing to satisfy properties ξ and $\mathfrak{f}$ on this functional model.

LEMMA 3.58. *Let $\mathcal{M}$ be a Σ-model, $\Phi \subseteq \mathrm{cwff}_o(\Sigma)$, and $\sim$ be a congruence on $\mathcal{M}$. We have $\mathcal{M}/_{\sim} \models \Phi$ iff $\mathcal{M} \models \Phi$. Furthermore, if $* \in \{\eta, \xi, \mathfrak{f}, \mathfrak{b}\}$ and $\sim$ satisfies property $*$, then $\mathcal{M}/_{\sim}$ satisfies property $*$.*

PROOF. Let $\boldsymbol{A}_o \in \Phi$. Since $\boldsymbol{A}$ is closed, $\mathcal{M} \models \boldsymbol{A}$, iff $\upsilon(\mathcal{E}(\boldsymbol{A})) \equiv \mathtt{T}$, iff $\upsilon^{\sim}(\mathcal{E}^{\sim}(\boldsymbol{A})) \equiv \mathtt{T}$, iff $\mathcal{M}/_{\sim} \models \boldsymbol{A}$. So, $\mathcal{M} \models \Phi$ iff $\mathcal{M}/_{\sim} \models \Phi$.

Suppose $\sim$ satisfies property η. Let $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, and an assignment φ into $\mathcal{M}/_{\sim}$ be given. Let φ^* be a corresponding assignment into $\mathcal{M}$ (cf. Definition 3.33). Since $\sim$ satisfies property η, we know $\mathcal{E}_{\varphi^*}(\boldsymbol{A}) \sim \mathcal{E}_{\varphi^*}(\boldsymbol{A}{\downarrow_{\beta\eta}})$. Taking equivalence classes, we have $\mathcal{E}^{\sim}_{\varphi}(\boldsymbol{A}) \equiv \mathcal{E}^{\sim}_{\varphi}(\boldsymbol{A}{\downarrow_{\beta\eta}})$.

Suppose $\sim$ satisfies property ξ. Let $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, a variable X_α and an assignment φ into $\mathcal{M}/_{\sim}$ be given. Again, let φ^* be a corresponding assignment into $\mathcal{M}$. Suppose $\mathcal{E}^{\sim}_{\varphi,[\mathsf{A}/X]}(\boldsymbol{M}) \equiv \mathcal{E}^{\sim}_{\varphi,[\mathsf{A}/X]}(\boldsymbol{N})$ for every $\mathsf{A} \in \mathcal{D}^{\sim}_\alpha$. This means $\mathcal{E}_{\varphi^*,[\mathsf{A}^*/X]}(\boldsymbol{M}) \sim \mathcal{E}_{\varphi^*,[\mathsf{A}^*/X]}(\boldsymbol{N})$ for every $\mathsf{A} \in \mathcal{D}^{\sim}_\alpha$. For any $\mathsf{a} \in \mathcal{D}_\alpha$, using Lemma 3.31, we know

$$\mathcal{E}_{\varphi^*,[\mathsf{a}/X]}(\boldsymbol{M}) \sim \mathcal{E}_{\varphi^*,[\mathsf{A}^*/X]}(\boldsymbol{M}) \sim \mathcal{E}_{\varphi^*,[\mathsf{A}^*/X]}(\boldsymbol{N}) \sim \mathcal{E}_{\varphi^*,[\mathsf{a}/X]}(\boldsymbol{N})$$

where $\mathsf{A} \in \mathcal{D}^{\sim}_\alpha$ is the equivalence class of a. Since $\sim$ satisfies property ξ, we know that $\mathcal{E}_{\varphi^*}(\lambda X.\boldsymbol{M}) \sim \mathcal{E}_{\varphi^*}(\lambda X.\boldsymbol{N})$. Taking equivalence classes, we see that $\mathcal{E}^{\sim}_{\varphi}(\lambda X.\boldsymbol{M}) \equiv \mathcal{E}^{\sim}_{\varphi}(\lambda X.\boldsymbol{N})$.

If $\sim$ is functional (satisfies property $\mathfrak{f}$), we know $\mathcal{M}/_{\sim}$ is functional (satisfies property $\mathfrak{f}$) by Theorem 3.13.

Finally, if $\sim$ satisfies property $\mathfrak{b}$, then clearly $\mathcal{D}^{\sim}_o$ has only two elements. So, $\mathcal{M}/_{\sim}$ satisfies property $\mathfrak{b}$. ⊣

DEFINITION 3.59 (Congruence relation $\dot\sim$). Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ be a Σ-model. Let $\mathsf{q}^\alpha \in \mathcal{D}_{\alpha\to\alpha\to o}$ be $\mathcal{E}(\mathbf{Q}^\alpha)$, i.e., the interpretation of Leibniz equality at type α. We define $\mathsf{a} \dot\sim \mathsf{b}$ in $\mathcal{D}_\alpha$ iff $\upsilon(\mathsf{q}^\alpha @ \mathsf{a} @ \mathsf{b}) \equiv \mathtt{T}$.

Before checking $\dot\sim$ is a congruence, we first show that it is at least reflexive.

LEMMA 3.60. *Let $\mathcal{M}$ be a Σ-model. For each type α and $\mathsf{a} \in \mathcal{D}_\alpha$, we have $\mathsf{a} \dot\sim \mathsf{a}$.*

PROOF. We need to check $\upsilon(\mathcal{E}(\mathbf{Q}^\alpha)@\mathsf{a}@\mathsf{a}) \equiv \mathtt{T}$. Let X_α be a variable of type α and φ be some assignment with $\varphi(X) \equiv \mathsf{a}$. Let $\mathsf{r} := \mathcal{E}_\varphi(\lambda P_{\alpha\to o}.\neg(PX) \vee PX))$. For any $\mathsf{p} \in \mathcal{D}_{\alpha\to o}$, since $\mathcal{E}$ is an evaluation function, we have

$$\upsilon(\mathsf{r}@\mathsf{p}) \equiv \upsilon(\mathcal{E}_{\varphi,[\mathsf{p}/P]}(\neg(PX) \vee PX)).$$

As $\mathcal{M}$ is a Σ-model, we have $\upsilon(\mathcal{E}_{\varphi,[\mathsf{p}/P]}(\neg(PX) \vee PX)) \equiv \mathtt{T}$ since either

$$\upsilon(\mathcal{E}_{\varphi,[\mathsf{p}/P]}(PX)) \equiv \mathtt{T} \quad \text{or} \quad \upsilon(\mathcal{E}_{\varphi,[\mathsf{p}/P]}(\neg(PX))) \equiv \mathtt{T}.$$

So, again since $\mathcal{M}$ is a Σ-model, $\upsilon(\mathscr{E}(\Pi^{\alpha\to o})@\mathsf{r}) \equiv \mathtt{T}$. By the definitions of r and $\dot{=}^{\alpha}$, we have $\upsilon(\mathscr{E}_{\varphi}(X \dot{=}^{\alpha} X)) \equiv \mathtt{T}$. As $X \dot{=}^{\alpha} X$ is a β-reduct of $\mathbf{Q}^{\alpha}XX$, we have $\upsilon(\mathscr{E}_{\varphi}(\mathbf{Q}^{\alpha}XX)) \equiv \mathtt{T}$ as well. Using $\varphi(X) \equiv \mathsf{a}$, we see that $\upsilon(\mathscr{E}(\mathbf{Q}^{\alpha})@\mathsf{a}@\mathsf{a}) \equiv \mathtt{T}$. ⊣

In order to check that $\dot{\sim}$ is a congruence, it is useful to unwind the definitions to better characterize when $\mathsf{a} \dot{\sim} \mathsf{b}$ for $\mathsf{a}, \mathsf{b} \in \mathscr{D}_{\alpha}$.

LEMMA 3.61 (Properties of $\dot{\sim}$). *Let $\mathcal{M}$ be a Σ-model. For each type α and $\mathsf{a}, \mathsf{b} \in \mathscr{D}_{\alpha}$, the following are equivalent*:

(1) $\mathsf{a} \dot{\sim} \mathsf{b}$.
(2) *For all variables X_{α} and Y_{α} and assignments φ such that $\varphi(X) \equiv \mathsf{a}$ and $\varphi(Y) \equiv \mathsf{b}$, we have $\upsilon(\mathscr{E}_{\varphi}(X \dot{=}^{\alpha} Y)) \equiv \mathtt{T}$.*
(3) *For every $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$, $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \mathtt{T}$ implies $\upsilon(\mathsf{p}@\mathsf{b}) \equiv \mathtt{T}$.*
(4) *For every $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$, $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b})$.*

PROOF. At each type α, let $\mathsf{q}^{\alpha} \in \mathscr{D}_{\alpha\to\alpha\to o}$ be the interpretation $\mathscr{E}(\mathbf{Q}^{\alpha})$ of Leibniz equality. By definition, $\mathsf{a} \dot{\sim} \mathsf{b}$ iff $\upsilon(\mathsf{q}^{\alpha}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$.

To show (1) implies (2), suppose $\mathsf{a} \dot{\sim} \mathsf{b}$ and φ is an assignment with $\varphi(X_{\alpha}) \equiv \mathsf{a}$ and $\varphi(Y_{\alpha}) \equiv \mathsf{b}$. Since $\upsilon(\mathsf{q}^{\alpha}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$, we have $\upsilon(\mathscr{E}_{\varphi}(\mathbf{Q}^{\alpha}XY)) \equiv \mathtt{T}$. Since $\mathscr{E}$ respects β-equality (cf. Remark 3.19), we have $\upsilon(\mathscr{E}_{\varphi}(X \dot{=}^{\alpha} Y)) \equiv \mathtt{T}$.

To show (2) implies (3), suppose $\upsilon(\mathscr{E}_{\varphi}(X \dot{=}^{\alpha} Y)) \equiv \mathtt{T}$ whenever φ is an assignment with $\varphi(X) \equiv \mathsf{a}$ and $\varphi(Y) \equiv \mathsf{b}$. Let X and Y be particular distinct variables of type α and φ be any such assignment with $\varphi(X) \equiv \mathsf{a}$ and $\varphi(Y) \equiv \mathsf{b}$. Let $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$ with $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \mathtt{T}$ and a variable $P_{\alpha\to o}$ be given. By assumption, $\upsilon(\mathscr{E}_{\varphi}(\forall P_{\alpha\to o}.\neg(PX) \vee (PY))) \equiv \mathtt{T}$. Since $\upsilon(\mathscr{E}_{\varphi,[\mathsf{p}/P]}(PX)) \equiv \upsilon(\mathsf{p}@\mathsf{a}) \equiv \mathtt{T}$, we have $\upsilon(\mathsf{p}@\mathsf{b}) \equiv \upsilon(\mathscr{E}_{\varphi,[\mathsf{p}/P]}(PY)) \equiv \mathtt{T}$.

To show (3) implies (4), let $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$ be given. If $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \mathtt{T}$, then we have $\upsilon(\mathsf{p}@\mathsf{b}) \equiv \mathtt{T}$ by assumption. So, $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b})$ in this case. Otherwise, we must have $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \mathtt{F}$. Let $\mathsf{q} := \mathscr{E}_{\varphi}(\lambda X_{\alpha}.\neg(P_{\alpha\to o}X))$ where φ is some assignment with $\varphi(P) := \mathsf{p}$. Since $\mathcal{M}$ is a model, $\upsilon(\mathsf{q}@\mathsf{a}) \equiv \upsilon(\mathscr{E}(\neg)@(\mathsf{p}@\mathsf{a})) \equiv \mathtt{T}$. Applying the assumption to q, we have $\upsilon(\mathsf{q}@\mathsf{b}) \equiv \mathtt{T}$ and so $\upsilon(\mathscr{E}(\neg)@(\mathsf{p}@\mathsf{b})) \equiv \mathtt{T}$. Thus, $\upsilon(\mathsf{p}@\mathsf{b}) \equiv \mathtt{F}$ and $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b})$ in this case as well.

To show (4) implies (1), suppose $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b})$ for every $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$. In particular, this holds for $\mathsf{p} := \mathsf{q}^{\alpha}@\mathsf{a} \in \mathscr{D}_{\alpha\to o}$. Since $\upsilon(\mathsf{q}^{\alpha}@\mathsf{a}@\mathsf{a}) \equiv \mathtt{T}$ by Lemma 3.60, we must have $\upsilon(\mathsf{q}^{\alpha}@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$. That is, $\mathsf{a} \dot{\sim} \mathsf{b}$. ⊣

THEOREM 3.62 (Properties of $\mathcal{M}/_{\dot{\sim}}$). *Let $\mathcal{M}$ be a Σ-model. Then $\dot{\sim}$ is a congruence relation on the model $\mathcal{M}$ and $\mathcal{M}/_{\dot{\sim}}$ satisfies property $\mathfrak{q}$. Furthermore, if for every type α, $=^{\alpha} \in \Sigma_{\alpha}$ and $\upsilon(\mathscr{E}(=^{\alpha})@a@b) \equiv \mathtt{T}$ iff $a \dot{\sim} b$ for all $\mathsf{a}, \mathsf{b} \in \mathscr{D}_{\alpha}$, then $\mathcal{M}/_{\dot{\sim}}$ is a Σ-model with primitive equality.*

PROOF. We first verify that $\dot{\sim}$ is an equivalence relation on each $\mathscr{D}_{\alpha}$. Reflexivity was shown in Lemma 3.60. To check symmetry and transitivity we use condition (4) in Lemma 3.61. For symmetry, let $\mathsf{a} \dot{\sim} \mathsf{b}$ in $\mathscr{D}_{\alpha}$ and $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$ be given. So, $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b})$. Generalizing over p, we have $\mathsf{b} \dot{\sim} \mathsf{a}$. For transitivity, let $\mathsf{a} \dot{\sim} \mathsf{b}$ and $\mathsf{b} \dot{\sim} \mathsf{c}$ in $\mathscr{D}_{\alpha}$ and $\mathsf{p} \in \mathscr{D}_{\alpha\to o}$ be given. So, $\upsilon(\mathsf{p}@\mathsf{a}) \equiv \upsilon(\mathsf{p}@\mathsf{b}) \equiv \upsilon(\mathsf{p}@\mathsf{c})$. Generalizing over p, we have $\mathsf{a} \dot{\sim} \mathsf{c}$.

We next verify that $\dot{\sim}$ is a congruence. Suppose $\mathsf{f} \dot{\sim} \mathsf{g}$ in $\mathscr{D}_{\alpha\to\beta}$ and $\mathsf{a} \dot{\sim} \mathsf{b} \in \mathscr{D}_{\alpha}$. To show $\mathsf{f}@\mathsf{a} \dot{\sim} \mathsf{g}@\mathsf{b}$ we use condition (3) in Lemma 3.61. Let $\mathsf{p} \in \mathscr{D}_{\beta\to o}$ with $\upsilon(\mathsf{p}@(\mathsf{f}@\mathsf{a})) \equiv \mathtt{T}$ be given. Let φ be an assignment with $\varphi(P_{\beta\to o}) \equiv \mathsf{p}$, $\varphi(X_{\alpha}) \equiv \mathsf{a}$

and $\varphi(G_{\alpha\to\beta}) \equiv \mathsf{g}$ for variables P, X and G. We can use Lemma 3.61(3) with $\mathscr{E}_\varphi(\lambda F_{\alpha\to\beta}.(P(FX)))$ and $\mathsf{f} \dot{\sim} \mathsf{g}$ to verify that $\upsilon(\mathsf{p}@(\mathsf{g}@\mathsf{a})) \equiv \mathtt{T}$. Using Lemma 3.61(3) with $\mathscr{E}_\varphi(\lambda X_\alpha.(P(GX)))$ and $\mathsf{a} \dot{\sim} \mathsf{b}$ verifies $\upsilon(\mathsf{p}@(\mathsf{g}@\mathsf{b})) \equiv \mathtt{T}$. So, $\mathsf{f}@\mathsf{a} \dot{\sim} \mathsf{g}@\mathsf{b}$.

It remains to check that $\upsilon(\mathsf{a}) \equiv \upsilon(\mathsf{b})$ whenever $\mathsf{a} \dot{\sim} \mathsf{b}$ for $\mathsf{a}, \mathsf{b} \in \mathscr{D}_o$. Let $\mathsf{a} \dot{\sim} \mathsf{b}$ in $\mathscr{D}_o$ be given. Applying Lemma 3.61(4) to $\mathscr{E}(\lambda X_o.X) \in \mathscr{D}_{o\to o}$ we have $\upsilon(\mathsf{a}) \equiv \upsilon(\mathscr{E}(\lambda X_o.X)@\mathsf{a}) \equiv \upsilon(\mathscr{E}(\lambda X_o.X)@\mathsf{b}) \equiv \upsilon(\mathsf{b})$ as desired. So, $\dot{\sim}$ is a congruence relation on $\mathscr{M}$.

Now, we show $\mathscr{M}/_{\dot{\sim}}$ satisfies property $\mathfrak{q}$. At each type α, let $\mathsf{q}^\alpha \in \mathscr{D}_{\alpha\to\alpha\to o}$ be the interpretation $\mathscr{E}(\mathbf{Q}^\alpha)$ of Leibniz equality. To check property $\mathfrak{q}$, we show that $[\![\mathsf{q}^\alpha]\!]_{\dot{\sim}}$ is the appropriate object in $\mathscr{D}^{\dot{\sim}}_{\alpha\to\alpha\to o}$ for each $\alpha \in \mathscr{T}$. Let $\mathsf{a}, \mathsf{b} \in \mathscr{D}_\alpha$ be given. Note that $[\![\mathsf{a}]\!]_{\dot{\sim}} \equiv [\![\mathsf{b}]\!]_{\dot{\sim}}$ is equivalent to $\mathsf{a} \dot{\sim} \mathsf{b}$.

Also, $\upsilon^{\dot{\sim}}([\![\mathsf{q}^\alpha]\!]_{\dot{\sim}}@^{\dot{\sim}}[\![\mathsf{a}]\!]_{\dot{\sim}}@^{\dot{\sim}}[\![\mathsf{b}]\!]_{\dot{\sim}}) \equiv \mathtt{T}$ is equivalent to $\upsilon(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$. So, we need to show that $\upsilon(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$ if and only if $\mathsf{a} \dot{\sim} \mathsf{b}$. But this is precisely the definition of $\dot{\sim}$.

The statement for primitive equality follows immediately by Theorem 3.55. ⊣

Now, we know that when one takes a quotient of a model $\mathscr{M}$ by $\dot{\sim}$, one obtains a model satisfying property $\mathfrak{q}$. It is worthwhile to note the following relationship between $\dot{\sim}$ and property $\mathfrak{q}$.

THEOREM 3.63. *Let* $\mathscr{M} \equiv (\mathscr{D}, @, \mathscr{E}, \upsilon)$ *be a* Σ*-model. The following are equivalent*:

(1) $\mathscr{M}$ *satisfies property* $\mathfrak{q}$.
(2) *For any congruence* $\sim$ *on* $\mathscr{M}$, *type* α, *and* $\mathsf{a}, \mathsf{b} \in \mathscr{D}_\alpha$, $\mathsf{a} \sim \mathsf{b}$ *implies* $\mathsf{a} \equiv \mathsf{b}$.
(3) *For any type* α, *and* $\mathsf{a}, \mathsf{b} \in \mathscr{D}_\alpha$, $\mathsf{a} \dot{\sim} \mathsf{b}$ *implies* $\mathsf{a} \equiv \mathsf{b}$.
(4) *For any type* α, $\mathfrak{L}^\alpha_{=}(\mathscr{E}(\mathbf{Q}^\alpha))$ *holds for* υ.

PROOF. To show (1) implies (2), suppose $\mathscr{M}$ satisfies $\mathfrak{q}$, $\sim$ is a congruence on $\mathscr{M}$, and $\mathsf{a} \sim \mathsf{b}$ for $\mathsf{a}, \mathsf{b} \in \mathscr{D}_\alpha$. Let $\mathsf{q}^\alpha \in \mathscr{D}_{\alpha\to\alpha\to o}$ be the object at type α guaranteed to exist by property $\mathfrak{q}$. Since $\mathsf{a} \sim \mathsf{b}$, we have $(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{a}) \sim (\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b})$. By property $\mathfrak{q}$, we have $\upsilon(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{a}) \equiv \mathtt{T}$ (since $\mathsf{a} \equiv \mathsf{a}$). Since $\sim$ is a congruence on the model, we have $\upsilon(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$. By property $\mathfrak{q}$, this means $\mathsf{a} \equiv \mathsf{b}$.

Since $\dot{\sim}$ is a particular congruence on $\mathscr{M}$, we know (2) implies (3).

To show (3) implies (4), we need to show $\mathfrak{L}^\alpha_{=}(\mathscr{E}(\mathbf{Q}^\alpha))$ holds for each type α. By the definition of $\dot{\sim}$, for every $\mathsf{a}, \mathsf{b} \in \mathscr{D}_\alpha$ we have $\upsilon(\mathscr{E}(\mathbf{Q}^\alpha)@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$, if and only if $\mathsf{a} \dot{\sim} \mathsf{b}$, iff $\mathsf{a} \equiv \mathsf{b}$. The last equivalence holds by our assumption that $\mathsf{a} \dot{\sim} \mathsf{b}$ implies that $\mathsf{a} \equiv \mathsf{b}$, and by Lemma 3.60.

For each type α, $\mathfrak{L}^\alpha_{=}(\mathscr{E}(\mathbf{Q}^\alpha))$ implies $\mathscr{E}(\mathbf{Q}^\alpha)$ is the witness required to show property $\mathfrak{q}$. So, we know (4) implies (1). ⊣

REMARK 3.64 (Congruences for Σ-models with primitive equality). Theorem 3.63 shows that once we have a model $\mathscr{M}$ which satisfies property $\mathfrak{q}$, there are no nontrivial congruences on $\mathscr{M}$. Hence, there are no nontrivial quotients of $\mathscr{M}$. In particular, the only possible congruence for a Σ-model with primitive equality is the trivial congruence given by the identity relation $\equiv$. Consequently, the quotient construction in the case of a Σ-model with primitive equality leads to essentially the same model again. We therefore do not consider quotients of models with primitive equality.

3.4. Σ-models over frames. In this section, we define the notion of an isomorphism between two models and show every functional Σ-model is isomorphic to a model over a frame. In particular, this shows that the model class $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ is simply the closure of the class $\mathfrak{H}$ of Henkin models under isomorphism of Σ-models.

DEFINITION 3.65 (Σ-model homomorphism/isomorphism). Let $\mathcal{M}^1 \equiv (\mathcal{D}^1, @^1, \mathcal{E}^1, \upsilon^1)$ and $\mathcal{M}^2 \equiv (\mathcal{D}^2, @^2, \mathcal{E}^2, \upsilon^2)$ be Σ-models. A *homomorphism* from $\mathcal{M}^1$ to $\mathcal{M}^2$ is a typed function $\kappa\colon \mathcal{D}^1 \longrightarrow \mathcal{D}^2$ such that κ is a homomorphism from the evaluation $(\mathcal{D}^1, @^1, \mathcal{E}^1)$ to the evaluation $(\mathcal{D}^2, @^2, \mathcal{E}^2)$ and $\upsilon^1(\mathsf{a}) \equiv \upsilon^2(\kappa(\mathsf{a}))$ for every $\mathsf{a} \in \mathcal{D}^1_o$.

A homomorphism i from $\mathcal{M}^1$ to $\mathcal{M}^2$ is called an *isomorphism* iff there is a homomorphism j from $\mathcal{M}^2$ to $\mathcal{M}^1$ where $j_\alpha\colon \mathcal{D}^2_\alpha \longrightarrow \mathcal{D}^1_\alpha$ is the inverse of $i_\alpha\colon \mathcal{D}^1_\alpha \longrightarrow \mathcal{D}^2_\alpha$ at each type α. Two models are said to be *isomorphic* if there is such an isomorphism. (It is clear from the definition that this is a symmetric relationship between models.)

REMARK 3.66. The class $\mathfrak{H}$ of Henkin models is not closed under isomorphism of models. Neither is the class $\mathfrak{ST}$ of standard models. This is because Henkin and standard models require that the domains $\mathcal{D}_{\alpha\to\beta}$ consist of functions from $\mathcal{F}(\mathcal{D}_\alpha; \mathcal{D}_\beta)$. We may, however, take a given Henkin model and appropriately modify it to obtain an isomorphic model that is not in the class of Henkin models. For example, we may choose $\mathcal{D}'_{\alpha\to\beta} := \{\, (0, f) \mid f \in \mathcal{D}_{\alpha\to\beta} \,\}$ and define @ appropriately (cf. Example 5.6 for a similar construction).

LEMMA 3.67. *Let $\mathcal{M}^1$ and $\mathcal{M}^2$ be isomorphic Σ-models.*

(1) *For any set of sentences Φ, $\mathcal{M}^1 \models \Phi$, iff $\mathcal{M}^2 \models \Phi$.*
(2) *If $\mathcal{M}^1$ is a Σ-model with primitive equality, then $\mathcal{M}^2$ is a Σ-model with primitive equality.*
(3) *If $* \in \{\mathsf{q}, \eta, \xi, \mathfrak{f}, \mathfrak{b}\}$ and $\mathcal{M}^1$ satisfies $*$, then $\mathcal{M}^2$ satisfies $*$.*

In particular, each model class $\mathfrak{M}_*$ is closed under isomorphism of models.

PROOF. Let i be a homomorphism from $\mathcal{M}^1 \equiv (\mathcal{D}^1, @^1, \mathcal{E}^1, \upsilon^1)$ to $\mathcal{M}^2 \equiv (\mathcal{D}^2, @^2, \mathcal{E}^2, \upsilon^2)$ and j be its inverse.

Let Φ be a set of sentences with $\mathcal{M}^1 \models \Phi$. That is, for every $\mathbf{A} \in \Phi$, $\upsilon^1(\mathcal{E}^1(\mathbf{A})) \equiv \mathtt{T}$. So, for every $\mathbf{A} \in \Phi$, $\upsilon^2(\mathcal{E}^2(\mathbf{A})) \equiv \upsilon^1(j(\mathcal{E}^2(\mathbf{A}))) \equiv \upsilon^1(\mathcal{E}^1(\mathbf{A})) \equiv \mathtt{T}$ (since $\mathbf{A}$ is closed, we can ignore the variable assignment). This shows $\mathcal{M}^2 \models \Phi$; the other direction is obtained by switching indices.

Suppose $\mathsf{q}^\alpha \in \mathcal{D}^1_{\alpha\to\alpha\to o}$ is such that $\mathfrak{L}^\alpha_=(\mathsf{q}^\alpha)$ holds for υ^1. We show that $\mathfrak{L}^\alpha_=(i(\mathsf{q}^\alpha))$ holds for υ^2. Given $\mathsf{a}, \mathsf{b} \in \mathcal{D}^2_\alpha$. We have $\mathsf{a} \equiv \mathsf{b}$, iff $j(\mathsf{a}) \equiv j(\mathsf{b})$, iff $\upsilon^1(\mathsf{q}^\alpha @^1 j(\mathsf{a}) @^1 j(\mathsf{b})) \equiv \mathtt{T}$, iff $\upsilon^2(i(\mathsf{q}^\alpha @^1 j(\mathsf{a}) @^1 j(\mathsf{b}))) \equiv \mathtt{T}$, iff $\upsilon^2(i(\mathsf{q}^\alpha) @^2 \mathsf{a} @^2 \mathsf{b})) \equiv \mathtt{T}$.

In particular, suppose $\mathcal{M}^1$ is a Σ-model with primitive equality. Then, we have $\mathfrak{L}^\alpha_=(\mathcal{E}^1(=^\alpha))$ for υ^1 at each type α. So, $\mathfrak{L}^\alpha_=(i(\mathcal{E}^1(=^\alpha)))$ holds for υ^2 at each type α. Since $i(\mathcal{E}^1(=^\alpha)) \equiv \mathcal{E}^2(=^\alpha)$, we know $\mathcal{M}^2$ is a Σ-model with primitive equality.

Next, suppose $\mathcal{M}^1$ satisfies property q. Let α be a type and q^α be the witness for property q in $\mathcal{M}^1$ at α. That is, $\mathfrak{L}^\alpha_=(\mathsf{q}^\alpha)$ holds for υ^1. We have shown $\mathfrak{L}^\alpha_=(i(\mathsf{q}^\alpha))$ holds for υ^2. Hence, $\mathcal{M}^2$ satisfies property q.

Suppose $\mathcal{M}^1$ satisfies property η. To show $\mathcal{M}^2$ satisfies η, let $\mathbf{A} \in \mathrm{wff}_\alpha(\Sigma)$ and an assignment φ into $\mathcal{M}^2$ be given. We compute

$$\begin{aligned}\mathcal{E}^2_\varphi(\mathbf{A}) &\equiv (i \circ j)(\mathcal{E}^2_\varphi(\mathbf{A})) \equiv i(\mathcal{E}^1_{j\circ\varphi}(\mathbf{A}))\\ &\equiv i(\mathcal{E}^1_{j\circ\varphi}(\mathbf{A}{\downarrow_{\beta\eta}})) \equiv (i \circ j)(\mathcal{E}^2_\varphi(\mathbf{A}{\downarrow_{\beta\eta}})) \equiv \mathcal{E}^2_\varphi(\mathbf{A}{\downarrow_{\beta\eta}}).\end{aligned}$$

So, $\mathcal{M}^2$ satisfies property η.

$\mathcal{M}^2$ satisfies ξ, let $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, a variable X_α, and an assignment ψ into $\mathcal{M}^2$ be given. Suppose $\mathcal{E}^2_{\psi,[\mathsf{b}/X]}(\boldsymbol{M}) \equiv \mathcal{E}^2_{\psi,[\mathsf{b}/X]}(\boldsymbol{N})$ for all $\mathsf{b} \in \mathcal{D}^2_\alpha$. For any $\mathsf{a} \in \mathcal{D}^1_\alpha$, we compute

$$\begin{aligned}\mathcal{E}^1_{j\circ\psi,[\mathsf{a}/X]}(\boldsymbol{M}) &\equiv j(\mathcal{E}^2_{i\circ j\circ\psi,[i(\mathsf{a})/X]}(\boldsymbol{M})) \equiv j(\mathcal{E}^2_{\psi,[i(\mathsf{a})/X]}(\boldsymbol{M}))\\ &\equiv j(\mathcal{E}^2_{\psi,[i(\mathsf{a})/X]}(\boldsymbol{N})) \equiv \mathcal{E}^1_{j\circ\psi,[\mathsf{a}/X]}(\boldsymbol{N}).\end{aligned}$$

Since $\mathcal{M}^1$ satisfies property ξ, we know $\mathcal{E}^1_{j\circ\psi}(\lambda X.\boldsymbol{M}) \equiv \mathcal{E}^1_{j\circ\psi}(\lambda X.\boldsymbol{N})$. Finally, we compute

$$\mathcal{E}^2_\psi(\lambda X.\boldsymbol{M}) \equiv i(\mathcal{E}^1_{j\circ\psi}(\lambda X.\boldsymbol{M})) \equiv i(\mathcal{E}^1_{j\circ\psi}(\lambda X.\boldsymbol{N})) \equiv \mathcal{E}^2_\psi(\lambda X.\boldsymbol{N}).$$

So, $\mathcal{M}^2$ satisfies property ξ.

Suppose $\mathcal{M}^1$ satisfies property $\mathfrak{f}$ and we are given $\mathsf{f}, \mathsf{g} \in \mathcal{D}^2_{\alpha\to\beta}$ for types α and β. Suppose further that $\mathsf{f}@^2\mathsf{b} \equiv \mathsf{g}@^2\mathsf{b}$ for every $\mathsf{b} \in \mathcal{D}^2_\alpha$. It is enough to show $j(\mathsf{f}) \equiv j(\mathsf{g})$. This follows from property $\mathfrak{f}$ in $\mathcal{M}^1$ if we can show $j(\mathsf{f})@^1\mathsf{a} \equiv j(\mathsf{g})@^1\mathsf{a}$ for every $\mathsf{a} \in \mathcal{D}^1_\alpha$. So, let $\mathsf{a} \in \mathcal{D}^1_\alpha$ be given. We finish the proof by computing

$$\begin{aligned}j(\mathsf{f})@^1\mathsf{a} &\equiv j(\mathsf{f})@^1(j\circ i)(\mathsf{a}) \equiv j(\mathsf{f}@^2 i(\mathsf{a}))\\ &\equiv j(\mathsf{g}@^2 i(\mathsf{a})) \equiv j(\mathsf{g})@^1(j\circ i)(\mathsf{a}) \equiv j(\mathsf{g})@^1\mathsf{a}.\end{aligned}$$

Finally, if $\mathcal{M}^1$ satisfies property $\mathfrak{b}$, then $\mathcal{D}^1_o$ has two elements. Since $i_o : \mathcal{D}^1_o \longrightarrow \mathcal{D}^2_o$ has inverse j_o, $\mathcal{D}^2_o$ must also have two elements. Thus, $\mathcal{M}^2$ satisfies property $\mathfrak{b}$. $\dashv$

THEOREM 3.68 (Models over frames). *Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ be a Σ-model which satisfies property $\mathfrak{f}$ (i.e., $\mathcal{M}$ is functional). Then there is an isomorphic model $\mathcal{M}^{fr}$ over a frame.*

PROOF. We define the model $\mathcal{M}^{fr} := (\mathcal{D}^{fr}, @^{fr}, \mathcal{E}^{fr}, \upsilon^{fr})$ by defining its components.

We first define the domains $\mathcal{D}^{fr}$ for $\mathcal{M}^{fr}$ by induction on types. We simultaneously define functions $i_\alpha : \mathcal{D}_\alpha \longrightarrow \mathcal{D}^{fr}_\alpha$ and $j_\alpha : \mathcal{D}^{fr}_\alpha \longrightarrow \mathcal{D}_\alpha$ which will witness that the two models are isomorphic. At each step of the definition, we check that i_α and j_α are mutual inverses. For base types $\alpha \in \{\iota, o\}$ let $\mathcal{D}^{fr}_\alpha := \mathcal{D}_\alpha$ and i_α and j_α be the identity functions (clearly mutual inverses).

Given two types α and β, we assume we have $\mathcal{D}^{fr}_\alpha$, mutual inverses $i_\alpha : \mathcal{D}_\alpha \to \mathcal{D}^{fr}_\alpha$ and $j_\alpha : \mathcal{D}^{fr}_\alpha \longrightarrow \mathcal{D}_\alpha$, as well as $\mathcal{D}^{fr}_\beta$ and mutual inverses $i_\beta : \mathcal{D}_\beta \to \mathcal{D}^{fr}_\beta$ and $j_\beta : \mathcal{D}^{fr}_\beta \longrightarrow \mathcal{D}_\beta$. We define

$$\mathcal{D}^{fr}_{\alpha\to\beta} := \left\{ f : \mathcal{D}^{fr}_\alpha \longrightarrow \mathcal{D}^{fr}_\beta \;\middle|\; \exists \mathsf{f} \in \mathcal{D}_{\alpha\to\beta}.\forall a \in \mathcal{D}^{fr}_\alpha.f(a) \equiv i_\beta(\mathsf{f}@j_\alpha(a)) \right\}.$$

Note that $\mathcal{D}^{fr}_{\alpha\to\beta} \subseteq \mathcal{F}(\mathcal{D}^{fr}_\alpha; \mathcal{D}^{fr}_\beta)$. To define the map $i_{\alpha\to\beta} : \mathcal{D}_{\alpha\to\beta} \longrightarrow \mathcal{D}^{fr}_{\alpha\to\beta}$, we let $i_{\alpha\to\beta}(\mathsf{f})$ be the function taking each $a \in \mathcal{D}^{fr}_\alpha$ to $i_\beta(\mathsf{f}@j_\alpha(a))$. This choice for $i_{\alpha\to\beta}(\mathsf{f})$ is clearly in $\mathcal{D}^{fr}_{\alpha\to\beta}$ by definition. To define the inverse map $j_{\alpha\to\beta} : \mathcal{D}^{fr}_{\alpha\to\beta} \longrightarrow \mathcal{D}_{\alpha\to\beta}$, we must use the fact that $\mathcal{M}$ is functional. Given any $f \in \mathcal{D}^{fr}_{\alpha\to\beta}$, by definition there is some $\mathsf{f} \in \mathcal{D}_{\alpha\to\beta}$ such that $f(a) \equiv i_\beta(\mathsf{f}@j_\alpha(a))$ for every $a \in \mathcal{D}^{fr}_\alpha$. (Note that the function f and object f are different in general.) By functionality and the fact that the i and j at types α and β are already inverses, this f is unique, since if

$i_\beta(\mathsf{f}@j_\alpha(a)) \equiv i_\beta(\mathsf{g}@j_\alpha(a))$ for every $a \in \mathscr{D}_\alpha^{fr}$, then $\mathsf{f}@j_\alpha(i_\alpha(\mathsf{a})) \equiv \mathsf{g}@j_\alpha(i_\alpha(\mathsf{a}))$ for every $\mathsf{a} \in \mathscr{D}_\alpha^{fr}$. That is, $\mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a}$ for every $\mathsf{a} \in \mathscr{D}_\alpha^{fr}$. So, for every $f \in \mathscr{D}_{\alpha\to\beta}^{fr}$, we define $j_{\alpha\to\beta}(f)$ to be the *unique* f such that $f(a) \equiv i_\beta(\mathsf{f}@j_\alpha(a))$. It is easy to check that $i_{\alpha\to\beta}$ and $j_{\alpha\to\beta}$ are mutually inverse.

For the applicative structure $(\mathscr{D}^{fr}, @^{fr})$ to be a frame, we are forced to let the application operator $@^{fr}$ to be function application. That is, for every $f \in \mathscr{D}_{\alpha\to\beta}^{fr}$ and $a \in \mathscr{D}_\alpha^{fr}$, $f@^{fr}a := f(a)$. We define the evaluation function $\mathscr{E}^{fr}$ simply by $\mathscr{E}_\varphi^{fr}(\boldsymbol{A}) := i(\mathscr{E}_{j\circ\varphi}(\boldsymbol{A}))$ for every $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$ and assignment φ into the applicative structure $(\mathscr{D}^{fr}, @^{fr})$. Since $\mathscr{D}_o^{fr} \equiv \mathscr{D}_o$, we can let $\upsilon^{fr} := \upsilon$.

We only sketch the remainder of the proof. First one can show that i and j preserve application. One can use this fact to verify that $\mathscr{E}^{fr}$ is an evaluation function so that $(\mathscr{D}^{fr}, @^{fr}, \mathscr{E}^{fr})$ is a Σ-evaluation, and that $\upsilon^{fr} \equiv \upsilon$ is a valuation function for this evaluation. This verifies $\mathscr{M}^{fr}$ is a model. Finally, to verify one has an isomorphism, one can easily check the remainder of the conditions for i and j to be homomorphisms between the models. These are isomorphisms since they are mutually inverse on the domains of each type. ⊣

We can conclude that $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ is simply the closure of the class of $\mathfrak{H}$ of Henkin models under isomorphism. Given any $\mathscr{M} \in \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$, by Theorem 3.68, there is an isomorphic model $\mathscr{M}^{fr}$ over a frame. By Lemma 3.67, this model $\mathscr{M}^{fr}$ satisfies $\mathfrak{q}$, $\mathfrak{f}$, and $\mathfrak{b}$ (since $\mathscr{M}$ does). Also, if primitive equality is present in the signature, by the same lemma we know $\mathscr{M}^{fr}$ is a model with primitive equality. That is, $\mathscr{M}^{fr} \in \mathfrak{H}$.

§4. **Properties of model classes.** In this section we discuss some properties of the model classes introduced in section 3. Our interest is in the properties of Leibniz equality and primitive equality.

DEFINITION 4.1 (Extensionality for Leibniz equality). We call a formula of the form

$$\mathrm{EXT}_{\doteq}^{\alpha\to\beta} \quad := \quad \forall F_{\alpha\to\beta}\mathbf{.}\forall G_{\alpha\to\beta}\mathbf{.}(\forall X_\alpha\mathbf{.}FX \doteq^\beta GX) \Rightarrow F \doteq^{\alpha\to\beta} G$$

an *axiom of (strong) functional extensionality for Leibniz equality*, and refer to the set

$$\mathrm{EXT}_{\doteq}^{\to} := \{\, \mathrm{EXT}_{\doteq}^{\alpha\to\beta} \mid \alpha, \beta \in \mathscr{T} \,\}$$

as the *axioms of (strong) functional extensionality for Leibniz equality*. Note that $\mathrm{EXT}_{\doteq}^{\to}$ specifies functionality of the relation corresponding to Leibniz equality $\doteq$. We call the formula

$$\mathrm{EXT}_{\doteq}^{o} \quad := \quad \forall A_o\mathbf{.}\forall B_o\mathbf{.}(A \Leftrightarrow B) \Rightarrow A \doteq^o B$$

the *axiom of Boolean extensionality*. We call the set $\mathrm{EXT}_{\doteq}^{\to} \cup \{\mathrm{EXT}_{\doteq}^{o}\}$ the *axioms of (strong) extensionality for Leibniz equality*.

In Examples 5.4 to 5.8 below we give concrete models in which $\mathrm{EXT}_{\doteq}^{o}$ and $\mathrm{EXT}_{\doteq}^{\alpha\to\beta}$ fail in various ways. First, we prove relationships between properties $\mathfrak{q}$, $\mathfrak{b}$ and $\mathfrak{f}$ and the statements $\mathrm{EXT}_{\doteq}^{o}$ and $\mathrm{EXT}_{\doteq}^{\to}$.

LEMMA 4.2 (Leibniz equality in Σ-models). *Let* $\mathscr{M} := (\mathscr{D}, @, \mathscr{E}, \upsilon)$ *be a* Σ*-model,* φ *be an assignment,* $\alpha \in \mathscr{T}$, *and* $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_\alpha(\Sigma)$.

(1) *If* $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$, *then* $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})) \equiv \mathtt{T}$.
(2) *If* $\mathcal{M}$ *satisfies property* $\mathfrak{q}$ *and* $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})) \equiv \mathtt{T}$, *then* $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$.

PROOF. Let φ be any assignment into $\mathcal{M}$. For the first part, suppose $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$. Given $\mathsf{r} \in \mathcal{D}_{\alpha\to o}$, we have either $\upsilon(\mathsf{r}@\mathcal{E}_\varphi(\boldsymbol{A})) \equiv \upsilon(\mathsf{r}@\mathcal{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{F}$ or $\upsilon(\mathsf{r}@\mathcal{E}_\varphi(\boldsymbol{B})) \equiv \upsilon(\mathsf{r}@\mathcal{E}_\varphi(\boldsymbol{A})) \equiv \mathtt{T}$. In either case, for any variable $P_{\alpha\to o}$ not in $\mathrm{free}(\boldsymbol{A}) \cup \mathrm{free}(\boldsymbol{B})$, we have $\upsilon(\mathcal{E}_{\varphi,[\mathsf{r}/P]}(\neg(P\boldsymbol{A}) \vee P\boldsymbol{B})) \equiv \mathtt{T}$. So, we have $\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \equiv \mathtt{T}$.

To show the second part, suppose $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})) \equiv \mathtt{T}$. By property $\mathfrak{q}$, there is some $\mathsf{q}^\alpha \in \mathcal{D}_{\alpha\to\alpha\to o}$ such that for $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$ we have $\upsilon(\mathsf{q}^\alpha@\mathsf{a}@\mathsf{b}) \equiv \mathtt{T}$ iff $\mathsf{a} \equiv \mathsf{b}$. Let $\mathsf{r} \equiv \mathsf{q}^\alpha@\mathcal{E}_\varphi(\boldsymbol{A})$. From $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})) \equiv \mathtt{T}$, we obtain $\mathcal{E}_{\varphi,[\mathsf{r}/P]}(\neg P\boldsymbol{A} \vee P\boldsymbol{B}) \equiv \mathtt{T}$ (where $P_{\alpha\to o} \notin \mathrm{free}(\boldsymbol{A}) \cup \mathrm{free}(\boldsymbol{B})$). Since $\mathcal{E}_{\varphi,[\mathsf{r}/P]}(P\boldsymbol{A}) \equiv \mathsf{q}^\alpha@\mathcal{E}_\varphi(\boldsymbol{A})@\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathtt{T}$, we must have $\upsilon(\mathcal{E}_{\varphi,[\mathsf{r}/P]}(P\boldsymbol{B})) \equiv \mathtt{T}$. That is, $\upsilon(\mathsf{q}^\alpha@\mathcal{E}_\varphi(\boldsymbol{A})@\mathcal{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{T}$. By the choice of q^α, we have $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$. ⊣

THEOREM 4.3 (Extensionality in Σ-models). *Let* $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ *be a* Σ-*model.*
(1) *If* $\mathcal{M}$ *satisfies property* $\mathfrak{q}$ *but not property* $\mathfrak{f}$, *then* $\mathcal{M} \not\models \mathrm{EXT}^{\to}_{\doteq}$.
(2) *If* $\mathcal{M}$ *satisfies property* $\mathfrak{q}$ *but not property* $\mathfrak{b}$, *then* $\mathcal{M} \not\models \mathrm{EXT}^{o}_{\doteq}$.
(3) *If* $\mathcal{M}$ *satisfies properties* $\mathfrak{q}$ *and* $\mathfrak{f}$, *then* $\mathcal{M} \models \mathrm{EXT}^{\to}_{\doteq}$.
(4) *If* $\mathcal{M}$ *satisfies property* $\mathfrak{b}$, *then* $\mathcal{M} \models \mathrm{EXT}^{o}_{\doteq}$.

Thus we can characterize the different semantical structures with respect to Boolean and functional extensionality by the table in Figure 5.[7]

in	$\mathfrak{M}_\beta, \mathfrak{M}_{\beta\eta}, \mathfrak{M}_{\beta\xi}$		$\mathfrak{M}_{\beta\mathfrak{f}}$		$\mathfrak{M}_{\beta\mathfrak{b}}, \mathfrak{M}_{\beta\eta\mathfrak{b}}, \mathfrak{M}_{\beta\xi\mathfrak{b}}$		$\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$	
formula	valid?	by	valid?	by	valid?	by	valid?	by
$\mathrm{EXT}^{\to}_{\doteq}$	—	1.	+	3.	—	1.	+	3.
$\mathrm{EXT}^{o}_{\doteq}$	—	2.	—	2.	+	4.[7]	+	4.[7]

FIGURE 5. Extensionality in Σ-models.

PROOF. Suppose $\mathcal{M}$ satisfies property $\mathfrak{q}$ but does not satisfy property $\mathfrak{f}$. Then there must be types α and β and objects $\mathsf{f}, \mathsf{g} \in \mathcal{D}_{\alpha\to\beta}$ such that $\mathsf{f} \not\equiv \mathsf{g}$ but $\mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a}$ for every $\mathsf{a} \in \mathcal{D}_\alpha$. Let $F_{\alpha\to\beta}, G_{\alpha\to\beta} \in \mathcal{V}_{\alpha\to\beta}$ be distinct variables, $X_\alpha \in \mathcal{V}_\alpha$, and φ be any assignment with $\varphi(F) \equiv \mathsf{f}$ and $\varphi(G) \equiv \mathsf{g}$. For any $\mathsf{a} \in \mathcal{D}_\alpha$, $\mathsf{f}@\mathsf{a} \equiv \mathsf{g}@\mathsf{a}$ implies $\upsilon(\mathcal{E}_{\varphi,[\mathsf{a}/X]}(FX \doteq^\beta GX)) \equiv \mathtt{T}$ by Lemma 4.2(1). Using the fact that υ is a valuation, we have $\upsilon(\mathcal{E}_\varphi(\forall X.(FX \doteq^\beta GX))) \equiv \mathtt{T}$. On the other hand, since $\mathsf{f} \not\equiv \mathsf{g}$ and $\mathcal{M}$ satisfies property $\mathfrak{q}$, we have $\upsilon(\mathcal{E}_\varphi(F \doteq^{\alpha\to\beta} G)) \equiv \mathtt{F}$ by contraposition of Lemma 4.2(2). This implies $\mathcal{M} \not\models \mathrm{EXT}^{\alpha\to\beta}_{\doteq}$.

Suppose $\mathcal{M}$ satisfies property $\mathfrak{q}$ but does not satisfy property $\mathfrak{b}$. Then, there must be at least three elements in $\mathcal{D}_o$. Since υ maps into a two element set, there must be two distinct elements $\mathsf{a}, \mathsf{b} \in \mathcal{D}_o$ such that $\upsilon(\mathsf{a}) \equiv \upsilon(\mathsf{b})$. Let $A_o, B_o \in \mathcal{V}_o$ be distinct variables and φ be any assignment into $\mathcal{M}$ with $\varphi(A) \equiv \mathsf{a}$ and $\varphi(B) \equiv \mathsf{b}$. By Lemma 3.48, we know $\upsilon(\mathcal{E}_\varphi(A \Leftrightarrow B)) \equiv \mathtt{T}$. Since $\mathsf{a} \not\equiv \mathsf{b}$ and property $\mathfrak{q}$ holds,

[7] The cases in the figure corresponding to Theorem 4.3(4) are actually special cases. In Theorem 4.3(4), we can infer a model satisfies $\mathrm{EXT}^{o}_{\doteq}$ even if property $\mathfrak{q}$ does not hold. However, the models in $\mathfrak{M}_{\beta\mathfrak{b}}$, $\mathfrak{M}_{\beta\eta\mathfrak{b}}$, $\mathfrak{M}_{\beta\xi\mathfrak{b}}$ and $\mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$ do satisfy property $\mathfrak{q}$ by the definition of these model classes.

by contraposition of Lemma 4.2(2), we know $\upsilon(\mathcal{E}_\varphi(A \doteq^o B)) \equiv \mathtt{F}$. It follows that $\mathcal{M} \not\models \mathrm{EXT}^o_{\doteq}$.

Let φ be any assignment into $\mathcal{M}$. From $\upsilon(\mathcal{E}_\varphi(\forall X_\alpha\text{.}FX \doteq GX)) \equiv \mathtt{T}$ we know $\upsilon(\mathcal{E}_{\varphi,[\mathsf{a}/X]}(FX \doteq GX)) \equiv \mathtt{T}$ holds for all $\mathsf{a} \in \mathcal{D}_\alpha$. By Lemma 4.2(2) we can conclude that $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(FX) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(GX)$ for all $\mathsf{a} \in \mathcal{D}_\alpha$ and hence $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(F)@\mathcal{E}_{\varphi,[\mathsf{a}/X]}(X) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(G)@\mathcal{E}_{\varphi,[\mathsf{a}/X]}(X)$ for all $\mathsf{a} \in \mathcal{D}_\alpha$. That is, $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(F)@\mathsf{a} \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(G)@\mathsf{a}$ for all $\mathsf{a} \in \mathcal{D}_\alpha$. Since X does not occur free in F or G, by property $\mathfrak{f}$ and Definition 3.18(3) we obtain $\mathcal{E}_\varphi(F) \equiv \mathcal{E}_\varphi(G)$. This finally gives us that $\upsilon(\mathcal{E}_\varphi(F \doteq^{\alpha\to\beta} G)) \equiv \mathtt{T}$ with Lemma 4.2(1). It follows that $\mathcal{M} \models \mathrm{EXT}^{\alpha\to\beta}_{\doteq}$ and $\mathcal{M} \models \mathrm{EXT}^{\to}_{\doteq}$, since α and β were chosen arbitrarily. Note that we certainly need the assumption that $\mathcal{M}$ satisfies property $\mathfrak{q}$ (which is employed within the application of Lemma 4.2(2). As explained in Remark 3.52, there is a functional model in which property $\mathfrak{q}$ fails and $\mathrm{EXT}^{\iota\to\iota}_{\doteq}$ is not valid.

Let $A_o, B_o \in \mathcal{V}_o$ be distinct variables and φ be any assignment into $\mathcal{M}$. Since property $\mathfrak{b}$ holds, we can assume $\mathcal{D}_o \equiv \{\mathtt{T}, \mathtt{F}\}$ and υ is the identity function. Suppose $\upsilon(\mathcal{E}_\varphi(A \Leftrightarrow B)) \equiv \mathtt{T}$. By Lemma 3.48, we have $\mathcal{E}_\varphi(A) \equiv \upsilon(\mathcal{E}_\varphi(A)) \equiv \upsilon(\mathcal{E}_\varphi(B)) \equiv \mathcal{E}_\varphi(B)$. By Lemma 4.2(1), we have $\upsilon(\mathcal{E}_\varphi(A \doteq^o B)) \equiv \mathtt{T}$. It follows that $\mathcal{M} \models \mathrm{EXT}^o_{\doteq}$. ⊣

Remark 4.4 (Alternative definitions of equality). Leibniz equality is a very prominent way of defining equality in higher-order logic. However, there are alternative definitions such as (cf. [6, p. 203])

$$\ddot{=}^\alpha \;:= \lambda X_\alpha Y_\alpha\text{.}\forall Q_{\alpha\to\alpha\to o}\text{.}(\forall Z_\alpha\text{.}QZZ) \Rightarrow QXY.$$

An important question is whether an alternative definition of equality is equivalent to the Leibniz definition in particular model classes. As Remark 3.47 shows, this has to be carefully investigated for each equality definition and each model class in question. We can show that for all $\boldsymbol{A}_\alpha, \boldsymbol{B}_\alpha \in \mathrm{cwff}_\alpha(\Sigma)$ $\boldsymbol{A} \ddot{=} \boldsymbol{B}$ and $\boldsymbol{A} \doteq \boldsymbol{B}$ are equivalent modulo υ for all $\mathcal{M} \in \mathfrak{M}_\beta$ (and thus for all other model classes). That is, we can show $\upsilon(\mathcal{E}(\boldsymbol{A} \ddot{=}^\alpha \boldsymbol{B})) \equiv \upsilon(\mathcal{E}(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}))$. Note that this is weaker than showing $\mathcal{E}(\boldsymbol{A} \ddot{=}^\alpha \boldsymbol{B}) \equiv \mathcal{E}(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})$. The key idea is to reduce the definition of $\ddot{=}$ to $\doteq$ (and vice versa) by instantiating the universally quantified set variables Q and P appropriately. We may, for instance, show $\boldsymbol{A} \ddot{=}^\alpha \boldsymbol{B}$ implies $\boldsymbol{A} \doteq^\alpha \boldsymbol{B}$ by choosing the instantiation $[\lambda U_\alpha V_\alpha\text{.}\forall P_{\alpha\to o}\text{.}PU \Rightarrow PV]$ for Q and the converse by choosing the instantiation $[\lambda V_\alpha\text{.}\forall Q_{\alpha\to\alpha\to o}\text{.}(\forall Z_\alpha\text{.}QZZ) \Rightarrow QAV]$ for P. As a consequence the properties of Leibniz equality with respect to extensionality also apply to $\ddot{=}$.

Definition 4.5 (Extensionality for primitive equality). Analogous to the extensionality axioms for Leibniz equality, we can define the *axioms of strong (functional and Boolean) extensionality for primitive equality*:

$$\mathrm{EXT}^{\alpha\to\beta}_{=} \;:= \forall F_{\alpha\to\beta}\text{.}\forall G_{\alpha\to\beta}\text{.}(\forall X_\alpha\text{.}FX =^\beta GX) \Rightarrow F =^{\alpha\to\beta} G$$

$$\mathrm{EXT}^o_{=} \;:= \forall A_o\text{.}\forall B_o\text{.}(A \Leftrightarrow B) \Rightarrow A =^o B.$$

As before we refer to the set $\mathrm{EXT}^{\to}_{=} := \{\mathrm{EXT}^{\alpha\to\beta}_{=} \mid \alpha, \beta \in \mathcal{T}\}$ as the *axioms of (strong) functional extensionality for primitive equality*.

The following lemma shows that in a Σ-model with primitive equality for each $\alpha \in \mathcal{T}$ the denotations of $=^\alpha$ and $\doteq^\alpha$ are identical modulo υ.

Lemma 4.6 (Primitive and Leibniz equality). *If* $\mathcal{M} := (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_*$ *is a* Σ*-model with primitive equality where* $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$, *then we have* $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} =^\alpha \boldsymbol{B})) \equiv \upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}))$ *for all assignments* φ *into* $\mathcal{M}$, *types* $\alpha \in \mathcal{T}$, *and* $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_\alpha(\Sigma)$.

Proof. Since property $\mathfrak{q}$ holds for $\mathcal{M} \in \mathfrak{M}_*$, by Lemma 4.2 parts (1) and (2), we have $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} \doteq^\alpha \boldsymbol{B})) \equiv \mathtt{T}$ iff $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$. Since $\mathcal{M}$ is a Σ-model with primitive equality, we know $\mathcal{E}_\varphi(\boldsymbol{A}) \equiv \mathcal{E}_\varphi(\boldsymbol{B})$ is equivalent to $\upsilon(\mathcal{E}(=^\alpha)@\mathcal{E}_\varphi(\boldsymbol{A})@\mathcal{E}_\varphi(\boldsymbol{B})) \equiv \mathtt{T}$, and hence to $\upsilon(\mathcal{E}_\varphi(\boldsymbol{A} =^\alpha \boldsymbol{B})) \equiv \mathtt{T}$. ⊣

Remark 4.7. Lemma 4.6 implies that for all models in our model classes $\mathfrak{M}_*$ the extensionality axioms for primitive equality are equivalent to the corresponding extensionality axioms for Leibniz equality. Thus, the analysis for the Leibniz versions applies directly to the versions using primitive equality. Also, Lemma 4.6 reinforces that (provided property $\mathfrak{q}$ holds) we can indeed use Leibniz equality to treat equality as a defined notion (relative to models in $\mathfrak{M}_*$). Thus, we principally do not need to assume the constants $=^\alpha$ to be in our signature. The critical part in this choice is that for ensuring the correct meaning for $\boldsymbol{Q}^\alpha$ we have to require the existence of an object representing the identity relation for each type in each Σ-model (cf. [2] for a discussion in the context of Henkin models). This requirement is automatically met if we consider primitive equality. Hence it seems natural to treat equality as primitive.

Remark 4.8 (Properties η and ξ). We have shown, in the presence of property $\mathfrak{q}$, a model $\mathcal{M}$ satisfies property $\mathfrak{f}$ iff $\mathcal{M} \models \mathrm{EXT}^{\rightarrow}_{\doteq}$. Similarly, we have shown that property $\mathfrak{b}$ corresponds to a model satisfying $\mathrm{EXT}^{o}_{\doteq}$. A corresponding analysis can be done for properties η and ξ (cf. Definition 3.46). Assume $\mathcal{M}$ satisfies property $\mathfrak{q}$. Then, $\mathcal{M}$ satisfies property η iff $\mathcal{M} \models \boldsymbol{A} \doteq^\alpha (\boldsymbol{A}{\downarrow_{\beta\eta}})$ for every type α and closed formula $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)$. Also, $\mathcal{M}$ satisfies property ξ iff

$$\mathcal{M} \models \forall F_{\alpha\to\beta}\centerdot\forall G_{\alpha\to\beta}\centerdot(\forall X_\alpha\centerdot FX \doteq^\beta GX) \Rightarrow (\lambda X\centerdot FX) \doteq^{\alpha\to\beta} (\lambda X\centerdot GX)$$

for all types α and β.

§5. Example models.

§5. Example models. We now sketch the construction of models in the model classes $\mathfrak{M}_*$ to demonstrate concretely how properties for Boolean, strong and weak functional extensionality can fail. We need this to show that the inclusions (cf. Figure 1) of the model classes defined in Section 3 are proper, and we indeed need all of them.

We start with the simplest example of a Henkin model, which we will call the *singleton model*, since the domain of individuals is a singleton. Note that the underlying evaluation of this model is not the singleton evaluation from Example 3.26 since $\mathcal{D}_o$ has two elements. In this model, all forms of extensionality are valid.

Example 5.1 (Singleton model—$\mathcal{M}^{\beta\mathfrak{f}\mathfrak{b}} \in \mathfrak{ST} \subseteq \mathfrak{H} \subseteq \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$). Let $(\mathcal{D}, @)$ be the full frame with $\mathcal{D}_o := \{\mathtt{T}, \mathtt{F}\}$ and $\mathcal{D}_\iota := \{*\}$. One can easily define an evaluation function $\mathcal{E}$ for this frame by induction on terms, using functions to interpret λ-abstractions. The identity function $\upsilon \colon \mathcal{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ is a valuation, assuming the logical constants are interpreted in the standard way (including primitive equality, if present in Σ). So, $\mathcal{M}^{\beta\mathfrak{f}\mathfrak{b}} := (\mathcal{D}, @, \mathcal{E}, \upsilon)$ defines a model. This model clearly satisfies all our properties $\mathfrak{b}$, $\mathfrak{f}$ (hence η and ξ) and $\mathfrak{q}$ (since the frame is full). So, $\mathcal{M}^{\beta\mathfrak{fb}} \in \mathfrak{ST} \subseteq \mathfrak{H} \subseteq \mathfrak{M}_{\beta\mathfrak{fb}}$.

REMARK 5.2. In particular, all our model classes are non-empty. By parts (3) and (4) of Theorem 4.3, we have $\mathcal{M}^{\beta\mathfrak{fb}} \models \mathrm{EXT}^{o}_{\dot{=}}$ and $\mathcal{M}^{\beta\mathfrak{fb}} \models \mathrm{EXT}^{\rightarrow}_{\dot{=}}$.

We can use the singleton model $\mathcal{M}^{\beta\mathfrak{fb}}$ to construct another model which makes the importance of property $\mathfrak{q}$ clear.

REMARK 5.3. Let $\mathcal{M}^{\beta\mathfrak{fb}} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ as above and $\mathcal{TE}(\Sigma)^{\beta} \equiv (\mathcal{D}^{\beta}, @^{\beta}, \mathcal{E}^{\beta})$ be the β-term evaluation as defined in Definition 3.35. Let $\upsilon'\colon \mathcal{D}^{\beta}_{o} \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ be the function $\upsilon'(\boldsymbol{A}) := \upsilon(\mathcal{E}(\boldsymbol{A}))$ for every $\boldsymbol{A} \in \mathrm{cwff}_{o}(\Sigma)\big\downarrow_{\beta}$. One can show $\mathcal{M}' := (\mathcal{D}^{\beta}, @^{\beta}, \mathcal{E}^{\beta}, \upsilon')$ is a Σ-model such that $\mathcal{M}' \models \boldsymbol{A}$ iff $\mathcal{M}^{\beta\mathfrak{fb}} \models \boldsymbol{A}$ for every sentence $\boldsymbol{A}$. In particular, $\mathcal{M}' \models \mathrm{EXT}^{o}_{\dot{=}}$ and $\mathcal{M}' \models \mathrm{EXT}^{\rightarrow}_{\dot{=}}$.

Nevertheless, $\mathcal{M}'$ fails to satisfy properties $\mathfrak{q}$, $\mathfrak{b}$, η and $\mathfrak{f}$. Property $\mathfrak{b}$ does not hold since $\mathcal{D}^{\beta}_{o} \equiv \mathrm{cwff}_{o}(\Sigma)\big\downarrow_{\beta}$ is infinite. Property η does not hold since, for example,

$$\mathcal{E}^{\beta}(\lambda F_{\iota\to\iota} X_{\iota}.FX) \equiv \lambda F_{\iota\to\iota} X_{\iota}.FX \not\equiv \lambda F_{\iota\to\iota}.F \equiv \mathcal{E}^{\beta}(\lambda F_{\iota\to\iota}.F).$$

Property $\mathfrak{f}$ cannot hold since property η does not hold. (On the other hand, property ξ does hold since the underlying evaluation is a term evaluation.)

We know now by Theorem 4.3, either part (1) or part (2), that property $\mathfrak{q}$ must not hold. A concrete way to see that property $\mathfrak{q}$ fails is to consider two distinct constants $a_{\iota}, b_{\iota} \in \Sigma_{\iota}$. We must have $\mathcal{M}^{\beta\mathfrak{fb}} \models a \doteq^{\iota} b$ (since $\mathcal{D}_{\iota}$ has only one element), and so $\mathcal{M}' \models a \doteq^{\iota} b$. On the other hand a and b are distinct elements (as distinct β-normal forms) in $\mathcal{D}^{\beta}_{\iota}$.

The model $\mathcal{M}'$ shows that property $\mathfrak{q}$ is needed in the proofs of parts (1) and (2) of Theorem 4.3.

EXAMPLE 5.4 (Failure of $\mathfrak{b}$—$\mathcal{M}^{\beta\mathfrak{f}} \in \mathfrak{M}_{\beta\mathfrak{f}} \setminus \mathfrak{M}_{\beta\mathfrak{fb}}$). Let $(\mathcal{D}, @)$ be the full frame with $\mathcal{D}_{o} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and $\mathcal{D}_{\iota} = \{0, 1\}$. We define an evaluation function $\mathcal{E}$ for this frame by defining $\mathcal{E}(\neg)$, $\mathcal{E}(\vee)$, and $\mathcal{E}(\Pi^{\alpha})$ to be the functions given in the following table:

$\mathcal{E}(\neg)$	a	b	c
	c	c	a

$\mathcal{E}(\vee)$	a	b	c
a	a	a	a
b	a	a	a
c	a	a	c

$$\mathcal{E}(\Pi^{\alpha})@\mathsf{f} = \begin{cases} \mathsf{a}, & \text{if } \mathsf{f}@\mathsf{g} \in \{\mathsf{a}, \mathsf{b}\} \text{ for all } \mathsf{g} \in \mathcal{D}_{\alpha}, \\ \mathsf{c}, & \text{if } \mathsf{f}@\mathsf{g} = \mathsf{c} \text{ for some } \mathsf{g} \in \mathcal{D}_{\alpha}. \end{cases}$$

We can choose $\mathcal{E}(w)$ to be arbitrary for parameters $w \in \Sigma$. Since the applicative structure $(\mathcal{D}, @)$ is a frame, hence functional, this uniquely determines $\mathcal{E}$ on all formulae. Also, since the frame is full, we are guaranteed that there will be enough functions to interpret λ-abstractions.

Let the map $\upsilon\colon \mathcal{D}_{o} \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ be defined by $\upsilon(\mathsf{a}) := \mathtt{T}$, $\upsilon(\mathsf{b}) := \mathtt{T}$ and $\upsilon(\mathsf{c}) := \mathtt{F}$. It is easy to check that $\mathcal{M}^{\beta\mathfrak{f}} := (\mathcal{D}, @, \mathcal{E}, \upsilon)$ is indeed a Σ-model. Since this is a model over a frame, we automatically know it satisfies property $\mathfrak{f}$. Since the frame is full, we know property $\mathfrak{q}$ holds. (By the same argument, if primitive equality is in the signature, we can ensure $\mathcal{E}(=^{\alpha})$ is interpreted appropriately for each type

α.) Clearly property $\mathfrak{b}$ fails, so we have $\mathcal{M}^{\beta\mathfrak{f}} \in \mathfrak{M}_{\beta\mathfrak{f}} \setminus \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$. By Theorem 4.3(2), $\mathcal{M}^{\beta\mathfrak{f}} \not\models \mathrm{EXT}^{o}_{\doteq}$.

In this model one can easily verify, if $\mathsf{d} := \mathcal{E}_\varphi(\boldsymbol{D}_o)$ and $\mathsf{e} := \mathcal{E}_\varphi(\boldsymbol{E}_o)$, then the values $\mathcal{E}_\varphi(\boldsymbol{D} \wedge \boldsymbol{E})$, $\mathcal{E}_\varphi(\boldsymbol{D} \Rightarrow \boldsymbol{E})$, and $\mathcal{E}_\varphi(\boldsymbol{D} \Leftrightarrow \boldsymbol{E})$ are given by the following tables:

		e:		
$\mathcal{E}(\boldsymbol{D} \wedge \boldsymbol{E})$		a	b	c
d:	a	a	a	c
	b	a	a	c
	c	c	c	c

		e:		
$\mathcal{E}(\boldsymbol{D} \Rightarrow \boldsymbol{E})$		a	b	c
d:	a	a	a	c
	b	a	a	c
	c	a	a	a

		e:		
$\mathcal{E}(\boldsymbol{D} \Leftrightarrow \boldsymbol{E})$		a	b	c
d:	a	a	a	c
	b	a	a	c
	c	c	c	a

Note that one can properly model the woodchuck / groundhog example from [39] referred to in the introduction in $\mathcal{M}^{\beta\mathfrak{f}}$.

EXAMPLE 5.5 (Groundhogs and woodchucks). Let $\mathcal{M}^{\beta\mathfrak{f}}$ be given as above and suppose $\mathrm{woodchuck}_{\iota\to o}$, $\mathrm{groundhog}_{\iota\to o}$, john_ι, and phil_ι are in the signature Σ. Let $\mathcal{E}(\mathrm{phil}) := 0$ and $\mathcal{E}(\mathrm{john}) := 1$. Let $\mathcal{E}(\mathrm{woodchuck})$ be the function $\mathsf{w} \in \mathcal{D}_{\iota\to o}$ with $\mathsf{w}(0) \equiv \mathsf{b}$ and $\mathsf{w}(1) \equiv \mathsf{c}$. Let $\mathcal{E}(\mathrm{groundhog})$ be the function $\mathsf{g} \in \mathcal{D}_{\iota\to o}$ with $\mathsf{g}(0) \equiv \mathsf{a}$ and $\mathsf{g}(1) \equiv \mathsf{c}$. One can show that the sentence $\forall X_\iota.(\mathrm{woodchuck}\, X) \Leftrightarrow (\mathrm{groundhog}\, X)$ is valid. Also, $\mathcal{E}(\mathrm{woodchuck\ phil}) \equiv \mathsf{b}$ and $\mathcal{E}(\mathrm{groundhog\ phil}) \equiv \mathsf{a}$, so the propositions (woodchuck phil) and (groundhog phil) are valid. Next, suppose $\mathrm{believe}_{\iota\to o\to o} \in \Sigma$ and $\mathcal{E}(\mathrm{believe})$ is the (Curried) function $\mathsf{bel} \in \mathcal{D}_{\iota\to o\to o}$ such that $\mathsf{bel}(1)(\mathsf{b}) \equiv \mathsf{b}$ and $\mathsf{bel}(1)(\mathsf{a}) \equiv \mathsf{bel}(1)(\mathsf{c}) \equiv \mathsf{bel}(0)(\mathsf{a}) \equiv \mathsf{bel}(0)(\mathsf{b}) \equiv \mathsf{bel}(0)(\mathsf{c}) \equiv \mathsf{c}$ (Intuitively, John believes propositions with value b, but not those with value a or c). So, believes john(woodchuck phil) is valid, while believes john(groundhog phil) is not.

As we have seen, Boolean extensionality fails when one has more than two values in $\mathcal{D}_o$. We can generalize the construction defining $\mathcal{D}_o := \{\mathtt{F}\} \cup \mathcal{B}$, where $\mathcal{B}$ is any set with $\mathtt{T} \in \mathcal{B}$ and $\mathtt{F} \notin \mathcal{B}$. The model will satisfy Boolean extensionality iff $\mathcal{B} \equiv \{\mathtt{T}\}$. In this way, we can easily construct models for the case with property $\mathfrak{b}$ and the case without property $\mathfrak{b}$ simultaneously. We will use this idea to parameterize the remaining model constructions by $\mathcal{B}$. These semantic constructions are similar to those in multi-valued logics, which have been studied for higher-order logic in [38]. In contrast to these logics where the logical connectives are adapted to talk about multiple truth values, in our setting we are mainly interested in multiple truth values as diverse υ-pre-images of $\mathtt{T}$ and $\mathtt{F}$.

EXAMPLE 5.6 (Failure of $\mathfrak{f}$ and η—$\mathcal{M}^{\beta\xi\mathfrak{b}} \in \mathfrak{M}_{\beta\xi\mathfrak{b}} \setminus \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$). We start by constructing a non-functional applicative structure by attaching distinguishing labels to functions without changing their applicative behavior. Let $\mathcal{B}$ be any set with $\mathtt{T} \in \mathcal{B}$ and $\mathtt{F} \notin \mathcal{B}$. Let $\mathcal{D}_o := \{\mathtt{F}\} \cup \mathcal{B}$ and $\mathcal{D}_\iota := \{*\}$ with $*$ as singleton element. For each function type $\alpha \to \beta$, let

$$\mathcal{D}_{\alpha\to\beta} := \{\, (i, f) \mid i \in \{0, 1\} \text{ and } f : \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta \,\}.$$

Technically, we should write $\mathcal{D}^{\mathcal{B}}$ for $\mathcal{D}$, but to ease the notation, we wait until the model is defined to make its dependence on $\mathcal{B}$ explicit. We define application by $(i, f)@\mathsf{a} := f(\mathsf{a})$ whenever $(i, f) \in \mathcal{D}_{\alpha\to\beta}$ and $\mathsf{a} \in \mathcal{D}_\alpha$. It is easy to see that $(\mathcal{D}, @)$ is an applicative structure and is not functional. Consider, for example, the unique function $u\colon \mathcal{D}_\iota \longrightarrow \mathcal{D}_\iota$. For both $(0,u),(1,u) \in \mathcal{D}_{\iota\to\iota}$ we have $(i,u)@* \equiv *$, although $(0,u) \not\equiv (1,u)$.

We can define an evaluation function by induction on terms. We must begin by interpreting the constants. For the logical constants, let $\mathcal{E}(\neg) := (0,n)$ where $n(\mathsf{b}) := \mathtt{F}$ for every $\mathsf{b} \in \mathcal{B}$ and $n(\mathtt{F}) := \mathtt{T}$. Let $\mathcal{E}(\vee) := (0,d)$ where $d(\mathsf{b}) := (0,k^{\mathtt{T}})$ for every $\mathsf{b} \in \mathcal{B}$, $d(\mathtt{F}) := (0,id)$, $k^{\mathtt{T}}$ is the constant $\mathtt{T}$ function and id is the identity function from $\mathcal{D}_o$ to $\mathcal{D}_o$. For each type α, let $d(\Pi^\alpha) := (0,\pi^\alpha)$ where for each $(i,f) \in \mathcal{D}_{\alpha\to o}$, $\pi^\alpha((i,f)) := \mathtt{T}$ if $f(\mathsf{a}) \in \mathcal{B}$ for all $\mathsf{a} \in \mathcal{D}_\alpha$ and $\pi^\alpha(i,f) := \mathtt{F}$ otherwise. For each type α, let $\mathsf{q}^\alpha := (0,q^\alpha) \in \mathcal{D}_{\alpha\to\alpha\to o}$ where $q^\alpha(\mathsf{a}) := (0,s^{\mathsf{a}})$ and $s^{\mathsf{a}}(\mathsf{b}) := \mathtt{T}$ if $\mathsf{a} \equiv \mathsf{b}$ and $s^{\mathsf{a}}(\mathsf{b}) := \mathtt{F}$ otherwise. If primitive equality is present in the signature, let $\mathcal{E}(=^\alpha) := \mathsf{q}^\alpha$. Let $\mathcal{E}(w) \in \mathcal{D}_\alpha$ be arbitrary for parameters $w \in \Sigma_\alpha$.

For variables, we must define $\mathcal{E}_\varphi(X) := \varphi(X)$. Similarly, for application, we must define $\mathcal{E}_\varphi(\boldsymbol{FA}) := \mathcal{E}_\varphi(\boldsymbol{F})@\mathcal{E}_\varphi(\boldsymbol{A})$. For λ-abstractions, we have a choice. To be definite, we choose $\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{B}_\beta) := (0,f)$ where $f\colon \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ is the function such that $f(\mathsf{a}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{B})$ for all $\mathsf{a} \in \mathcal{D}_\alpha$.

With some work (which we omit), one can show that this $\mathcal{E}$ is an evaluation function. Furthermore, taking υ to be the function such that $\upsilon(\mathsf{b}) := \mathtt{T}$ for every $\mathsf{b} \in \mathcal{B}$ and $\upsilon(\mathtt{F}) := \mathtt{F}$, one can easily show that this is a valuation. Hence, $\mathcal{M}^{\mathcal{B}} := (\mathcal{D}, @, \mathcal{E}, \upsilon)$ is a Σ-model.

The objects q^α witness property $\mathfrak{q}$ for $\mathcal{M}^{\mathcal{B}}$ (and also show that this is a model with primitive equality, when primitive equality is in the signature). Note that the objects $(1,q^\alpha)$ also witness property $\mathfrak{q}$. So, in the non-functional case such witnesses are not unique.

We have already noted that property $\mathfrak{f}$ fails, since the applicative structure is not functional. One may question whether properties η or ξ hold. In fact, property η does not, as one may verify by computing, for example, $\mathcal{E}(\lambda F_{\alpha\to\beta}.F)$ and $\mathcal{E}(\lambda F_{\alpha\to\beta}X_\alpha.FX)$ for types α and β. We have $\mathcal{E}(\lambda F_{\alpha\to\beta}.F) \equiv (0,id)$ where id is the identity function from $\mathcal{D}_{\alpha\to\beta}$ to $\mathcal{D}_{\alpha\to\beta}$. However, $\mathcal{E}(\lambda F_{\alpha\to\beta}X_\alpha.FX) \equiv (0,p)$ where p is the function from $\mathcal{D}_{\alpha\to\beta}$ to $\mathcal{D}_{\alpha\to\beta}$ such that $p((i,f)) \equiv (0,f)$ for each $f\colon \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$. Property ξ does hold.[8] The reason is that if $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$, then $\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{M}) \equiv (0,f) \equiv \mathcal{E}_\varphi(\lambda X.\boldsymbol{N})$ where $f(\mathsf{a}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$ for every $\mathsf{a} \in \mathcal{D}_\alpha$.

Since $\mathcal{M}^{\mathcal{B}}$ is satisfies property $\mathfrak{q}$ but not property $\mathfrak{f}$, by Theorem 4.3(1) we have $\mathcal{M}^{\mathcal{B}} \not\models \mathrm{EXT}^{\alpha\to\beta}_{\doteq}$ for some types α and β. (One can easily check that, in fact, $\mathcal{M}^{\mathcal{B}} \not\models \mathrm{EXT}^{\alpha\to\beta}_{\doteq}$ for all types α and β by considering the witnesses $(0,f)$ and $(1,f)$ in $\mathcal{D}_{\alpha\to\beta}$ where $f\colon \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ is any function.)

If $\mathcal{B} \equiv \{\mathtt{T}\}$, then the model $\mathcal{M}^{\beta\xi\mathfrak{b}} := \mathcal{M}^{\{\mathtt{T}\}}$ satisfies property $\mathfrak{b}$. So, we know $\mathcal{M}^{\beta\xi\mathfrak{b}} \in \mathfrak{M}_{\beta\xi\mathfrak{b}} \setminus \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$. On the other hand, if b is any value with $\mathsf{b} \notin \{\mathtt{T},\mathtt{F}\}$, and $\mathcal{B} \equiv \{\mathtt{T},\mathsf{b}\}$, then the model $\mathcal{M}^{\beta\xi} := \mathcal{M}^{\{\mathtt{T},\mathsf{b}\}}$ does not satisfy property $\mathfrak{b}$. In this case, we know $\mathcal{M}^{\beta\xi} \in \mathfrak{M}_{\beta\xi} \setminus (\mathfrak{M}_{\beta\mathfrak{f}} \cup \mathfrak{M}_{\beta\xi\mathfrak{b}})$.

[8] This construction is an example of how one constructs models for the simply typed λ-calculus using retractions. Such constructions will always yield models satisfying property ξ, but only yield models satisfying property η when each retraction is an isomorphism, in which case the applicative structure is functional.

REMARK 5.7. Let $\mathcal{M}^{\mathcal{B}}$ be the Σ-model $(\mathcal{D}, @, \mathcal{E}, \upsilon)$ constructed in Example 5.6. We can define an alternative evaluation function $\mathcal{E}'$ by induction on terms. For all $w \in \Sigma$, let $\mathcal{E}'(w) := \mathcal{E}(w)$. For variables, we define $\mathcal{E}'_\varphi(X) := \varphi(X)$. For application, we must define $\mathcal{E}'_\varphi(\boldsymbol{FA}) := \mathcal{E}'_\varphi(\boldsymbol{F})@\mathcal{E}'_\varphi(\boldsymbol{A})$. For λ-abstractions, we choose $\mathcal{E}'_\varphi(\lambda X_\alpha \boldsymbol{.} \boldsymbol{B}_\beta) := (1, f)$ where $f : \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ is the function such that $f(\mathsf{a}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{B})$ for all $\mathsf{a} \in \mathcal{D}_\alpha$. We omit checking $\mathcal{E}'$ is an evaluation function, but the verification is that same is checking $\mathcal{E}$ is an evaluation function. Notice that $\mathcal{E}$ and $\mathcal{E}'$ agree on all constants (by definition). However, they are different evaluation functions. For example,

$$\mathcal{E}(\lambda X_\iota \boldsymbol{.} X) \equiv (0, \mathrm{id}) \not\equiv (1, \mathrm{id}) \equiv \mathcal{E}'(\lambda X_\iota \boldsymbol{.} X)$$

where $\mathrm{id}\colon \mathcal{D}_\iota \longrightarrow \mathcal{D}_\iota$ is the identity function.This example shows that evaluation functions are not uniquely determined by their values on constants in non-functional models.

In Lemma 3.14, we have shown that $\beta\eta$-equality induces a functional congruence if the Σ_α is infinite for all types α. As a result, with such signatures, the term evaluation $\mathcal{TE}(\Sigma)^{\beta\eta}$ is functional (cf. Lemma 3.36). As noted in Remark 3.15, if Σ is finite, we cannot show that functionality holds. Nevertheless, even if Σ is finite, the evaluation $\mathcal{TE}(\Sigma)^{\beta\eta}$ interprets $\beta\eta$-convertible terms the same. We can use this idea to construct non-functional models which satisfy property η.

EXAMPLE 5.8 (Failure of ξ—Instances of $\mathfrak{M}_\beta$, $\mathfrak{M}_{\beta\eta}$, $\mathfrak{M}_{\beta\mathfrak{b}}$, $\mathfrak{M}_{\beta\eta\mathfrak{b}}$). Again, let $\mathcal{B}$ be any set with $\mathtt{T} \in \mathcal{B}$ and $\mathtt{F} \notin \mathcal{B}$. Choose constants $c_\iota, c_o \in \Sigma$ and let $\Sigma' := \{c_\iota, c_o\}$. By induction on types, we define $\boldsymbol{C}'_\alpha \in \mathrm{cwff}_\alpha(\Sigma')\big\downarrow_{\beta\eta} \subseteq \mathrm{cwff}_\alpha(\Sigma')\big\downarrow_\beta$. At base types, let $\boldsymbol{C}'_\iota := c_\iota$ and $\boldsymbol{C}'_o := c_o$. At function types, let $\boldsymbol{C}'_{\alpha\to\beta} := \lambda X_\alpha \boldsymbol{.} \boldsymbol{C}'_\beta$. (Thus each $\boldsymbol{C}'_\alpha$ is of the form $\lambda \overline{X} \boldsymbol{.} c_\beta$ where $\beta \in \{\iota, o\}$.) In particular, $\mathrm{cwff}_\alpha(\Sigma')\big\downarrow_{\beta\eta}$ and $\mathrm{cwff}_\alpha(\Sigma')\big\downarrow_\beta$ are non-empty for each type α.

We can now inductively define a map ρ from $\mathrm{wff}_\alpha(\Sigma)$ to $\mathrm{wff}_\alpha(\Sigma')$ which collapses terms to the smaller signature. For variables, let $\rho(X) := X$. For constants $w_\alpha \in \Sigma$ (including logical constants), let $\rho(w_\alpha) := \boldsymbol{C}'_\alpha$. For application and λ-abstraction, we simply use $\rho(\boldsymbol{FA}) := \rho(\boldsymbol{F})\rho(\boldsymbol{A})$ and $\rho(\lambda X \boldsymbol{.} \boldsymbol{A}) := \lambda X \boldsymbol{.} \rho(\boldsymbol{A})$. By induction on the formula $\boldsymbol{A}$, one can show $[\rho(\boldsymbol{B})/X]\rho(\boldsymbol{A}) \equiv \rho([\boldsymbol{B}/X]\boldsymbol{A})$ for any $\boldsymbol{A} \in \mathrm{wff}_\alpha(\Sigma)$, $\boldsymbol{B} \in \mathrm{wff}_\beta(\Sigma)$ and X_β. From this, one can show $\rho(\boldsymbol{A}) \equiv_{\beta\eta} \rho(\boldsymbol{B})$ whenever $\boldsymbol{A} \equiv_{\beta\eta} \boldsymbol{B}$ for every $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{wff}_\alpha(\Sigma)$. Note also that $\rho(\boldsymbol{A}') \equiv \boldsymbol{A}'$ for every $\boldsymbol{A}' \in \mathrm{wff}_\alpha(\Sigma')$.

We can construct a non-functional applicative structure using an indexing technique similar to Example 5.6. In this case, instead of indexing with $i \in \{0, 1\}$, we use terms in $\mathrm{cwff}_\alpha(\Sigma')\big\downarrow_*$ as indices. (Here $\boldsymbol{A}\big\downarrow_*$ means the β-normal form if $* \equiv \beta$ and the $\beta\eta$-normal form if $* \equiv \beta\eta$.) In essence, this index records some information about the "implementation" of the function. Note that $\mathrm{cwff}_\iota(\Sigma')\big\downarrow_* \equiv \{c_\iota\}$ and $\mathrm{cwff}_o(\Sigma')\big\downarrow_* \equiv \{c_o\}$. Let $\mathcal{D}_\iota := \{(c_\iota, 0)\}$ and $\mathcal{D}_o := \{(c_o, \mathtt{F})\} \cup \{(c_o, \mathsf{b}) \mid \mathsf{b} \in \mathcal{B}\}$. For function types, let $\mathcal{D}_{\alpha\to\beta}$ be the set of pairs $(\boldsymbol{F}'_{\alpha\to\beta}, f)$, where $\boldsymbol{F}' \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma')\big\downarrow_*$ and $f : \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ is any function such that $f(\boldsymbol{A}', a) \equiv ((\boldsymbol{F}'\boldsymbol{A}')\big\downarrow_*, b)$ for some value b. Application is defined as in Example 5.6: $(\boldsymbol{F}, f)@\mathsf{a} := f(\mathsf{a})$. The construction of this applicative structure closely follows Andrews' υ-complexes in [1], except we have a very restricted signature Σ' which does not include logical constants.

To show that each domain is non-empty, we construct a particular element $\mathsf{c}^\alpha \in \mathcal{D}_\alpha$ for each type α. (This element will also be used to interpret parameters.) Let $\mathsf{c}^\iota := (c_\iota, 0)$, $\mathsf{c}^o := (c_o, \mathtt{F})$, and $\mathsf{c}^{\alpha\to\beta} := (\boldsymbol{C}'_{\alpha\to\beta}, k)$ where $k\colon \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ is the constant function $k(\mathsf{a}) := \mathsf{c}^\beta$ for every $\mathsf{a} \in \mathcal{D}_\alpha$. The fact that $\mathsf{c}^{\alpha\to\beta} \in \mathcal{D}_{\alpha\to\beta}$ follows from $(\boldsymbol{C}'_{\alpha\to\beta}\boldsymbol{A})\Downarrow_* \equiv \boldsymbol{C}'_\beta$.

One can see that the applicative structure is non-functional by noting $(\lambda X_\iota.X, f)$ and $(\lambda X_\iota.c_\iota, f)$ are distinct members of $\mathcal{D}_{\iota\to\iota}$, where f is the unique function taking $\mathcal{D}_\iota$ into itself. However, $(\lambda X_\iota.X, f)@\mathsf{c}^\iota \equiv \mathsf{c}^\iota \equiv (\lambda X_\iota.c_\iota, f)@\mathsf{c}^\iota$. In fact, once we define the evaluation function, this same example will show that property ξ will fail.

Let $\upsilon\colon \mathcal{D}_o \longrightarrow \{\mathtt{T}, \mathtt{F}\}$ be $\upsilon((c_o, \mathtt{F})) := \mathtt{F}$ and $\upsilon((c_o, \mathsf{b})) := \mathtt{T}$ for each $\mathsf{b} \in \mathcal{B}$. This will be the valuation function on the model.

We only sketch the definition of the evaluation function $\mathcal{E}$ and the proof that this gives a model $\mathcal{M}^{*,\mathcal{B}} := (\mathcal{D}, @, \mathcal{E}, \upsilon)$. We can define $\mathcal{E}$ by induction on terms. First, we interpret parameters $w_\alpha \in \Sigma$ by $\mathcal{E}(w_\alpha) := \mathsf{c}^\alpha$. For logical constants $a_\alpha \in \Sigma$, we choose the first component of $\mathcal{E}(a_\alpha)$ to be $\boldsymbol{C}'_\alpha$ and the second component to be an appropriate function. We can define the witnesses q^α in a similar way and use these to interpret primitive equality, if it is present in the signature.

We are forced to let $\mathcal{E}_\varphi(X) := \varphi(X)$ and $\mathcal{E}_\varphi(\boldsymbol{FA}) := \mathcal{E}_\varphi(\boldsymbol{F})@\mathcal{E}_\varphi(\boldsymbol{A})$. For the λ-abstraction step, we choose $\mathcal{E}_\varphi(\lambda X_\alpha.\boldsymbol{B}_\beta) := ((\sigma(\rho(\lambda X.\boldsymbol{B})))\Downarrow_*, f)$, where $f : \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ satisfies $f(\mathsf{a}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{B})$ for all $\mathsf{a} \in \mathcal{D}_\alpha$ and σ is the substitution defined by letting $\sigma(Y)$ be the first component of $\varphi(Y)$ for each $Y \in \operatorname{free}(\lambda X.\boldsymbol{B})$. In order to show $\mathcal{E}$ is well-defined, one shows the first component of $\mathcal{E}_\varphi(\boldsymbol{A})$ is $(\sigma(\rho(\boldsymbol{A})))\Downarrow_*$ (where σ is the substitution for $\operatorname{free}(\boldsymbol{A})$ defined from the first components of the values of φ) for every formula $\boldsymbol{A}$.

The fact that $\mathcal{E}$ evaluates variables and application properly is immediate from the definition. The fact that $\mathcal{E}_\varphi(\boldsymbol{A})$ depends only the free variables in $\boldsymbol{A}$ follows by an induction on the definition of $\mathcal{E}$. To show $\mathcal{E}$ respects β-conversion if $* \equiv \beta$ and $\beta\eta$-conversion if $* \equiv \beta\eta$ (so that the model will also satisfy property η), one first shows $\mathcal{E}$ respects a single $\beta[\eta]$-reduction, then does an induction on the position of the redex, and finally does an induction on the number of $\beta[\eta]$-reductions.

Once these details are checked, we know $\mathcal{M}^{*,\mathcal{B}}$ is a model (with primitive equality, if present) satisfying property q. We already know the model will not satisfy property $\mathfrak{f}$ since the applicative structure is not functional. We can also check that the model will not satisfy property ξ by considering $\mathcal{E}(\lambda X_\iota.X)$ and $\mathcal{E}(\lambda X_\iota.c_\iota)$. We know $\mathcal{E}(\lambda X_\iota.X) \not\equiv \mathcal{E}(\lambda X_\iota.c_\iota)$ since the first components $((\lambda X_\iota.X)$ and $(\lambda X_\iota.c_\iota))$ are not equal. However, $\mathcal{D}_\iota$ has only one element, $\mathsf{c}^\iota \equiv (c_\iota, 0)$. So, we must have $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(X) \equiv \mathsf{c}^\iota \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(c_\iota)$ for every $\mathsf{a} \in \mathcal{D}_\iota$. This shows property ξ fails.

If $* \equiv \beta\eta$, then we have noted above that $\mathcal{E}$ respects $\beta\eta$-conversion. So, in this case, the model satisfies property η. If $* \equiv \beta$, then we can easily check $\mathcal{E}(\lambda F_{\iota\to\iota}X_\iota.FX) \not\equiv \mathcal{E}(\lambda F_{\iota\to\iota}.F)$ since the first components will differ. So, in this case, the model does not satisfy property η.

As in Example 5.6, if $\mathcal{B} \equiv \{\mathtt{T}\}$, then $\mathcal{M}^{\beta\mathfrak{b}} := \mathcal{M}^{\beta,\{\mathtt{T}\}}$ and $\mathcal{M}^{\beta\eta\mathfrak{b}} := \mathcal{M}^{\beta\eta,\{\mathtt{T}\}}$ satisfy property $\mathfrak{b}$. So, we know $\mathcal{M}^{\beta\mathfrak{b}} \in \mathfrak{M}_{\beta\mathfrak{b}} \setminus (\mathfrak{M}_{\beta\eta\mathfrak{b}} \cup \mathfrak{M}_{\beta\xi\mathfrak{b}})$ and $\mathcal{M}^{\beta\eta\mathfrak{b}} \in \mathfrak{M}_{\beta\eta\mathfrak{b}} \setminus \mathfrak{M}_{\beta\mathfrak{f}\mathfrak{b}}$. If $\mathcal{B} \equiv \{\mathtt{T}, \mathsf{b}\}$ where b is any value with $\mathsf{b} \notin \{\mathtt{T}, \mathtt{F}\}$, then the models $\mathcal{M}^\beta := \mathcal{M}^{\beta,\{\mathtt{T},\mathsf{b}\}}$ and $\mathcal{M}^{\beta\eta} := \mathcal{M}^{\beta\eta,\{\mathtt{T},\mathsf{b}\}}$ do not satisfy property $\mathfrak{b}$, so $\mathcal{M}^\beta \in \mathfrak{M}_\beta \setminus (\mathfrak{M}_{\beta\eta} \cup \mathfrak{M}_{\beta\xi} \cup \mathfrak{M}_{\beta\mathfrak{b}})$ and $\mathcal{M}^{\beta\eta} \in \mathfrak{M}_{\beta\eta} \setminus (\mathfrak{M}_{\beta\mathfrak{f}} \cup \mathfrak{M}_{\beta\eta\mathfrak{b}})$.

In particular, the models $\mathcal{M}^{\beta\eta}$ and $\mathcal{M}^{\beta\eta\mathfrak{b}}$ show that respecting η-conversion does not guarantee strong functional extensionality.

Thus we have given (sketches of) concrete models that distinguish model classes and shown that the inclusions between the $\mathfrak{M}_*$ model classes in Figure 1 are proper.

§6. Model existence. In this section we present the model existence theorems for the different semantical notions introduced in Section 3. The model existence theorems have the following form, where $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$:

THEOREM (Model existence). *For a given abstract consistency class $\Gamma_\Sigma \in \mathfrak{Acc}_*$ (cf. Definition 6.7) and a set $\Phi \in \Gamma_\Sigma$ there is a Σ-model $\mathcal{M}$ of Φ, such that $\mathcal{M} \in \mathfrak{M}_*$ (cf. Definition 3.49).*

The most important tools used in the proofs of the model existence theorems are the so-called Σ-Hintikka sets. These sets allow computations that resemble those in the considered semantical structures (e.g., Henkin models) and allow us to construct appropriate valuations for the term evaluation $\mathcal{TE}(\Sigma)^\beta$ defined in Definition 3.35. The key step in the proof of the model existence theorems is an extension lemma, which guarantees a Σ-Hintikka set $\mathcal{H}$ for any sufficiently Σ-pure set of sentences Φ in Γ_Σ.

6.1. Abstract consistency. Let us now review a few technicalities that we will need for the proofs of the model existence theorems.

DEFINITION 6.1 (Compactness). Let $\mathcal{C}$ be a class of sets.

(1) $\mathcal{C}$ is called *closed under subsets* if for any sets S and T, $S \in \mathcal{C}$ whenever $S \subseteq T$ and $T \in \mathcal{C}$.
(2) $\mathcal{C}$ is called *compact* if for every set S we have $S \in \mathcal{C}$ iff every finite subset of S is a member of $\mathcal{C}$.

LEMMA 6.2. *If $\mathcal{C}$ is compact, then $\mathcal{C}$ is closed under subsets.*

PROOF. Suppose $S \subseteq T$ and $T \in \mathcal{C}$. Every finite subset A of S is a finite subset of T, and since $\mathcal{C}$ is compact we know that $A \in \mathcal{C}$. Thus $S \in \mathcal{C}$. $\dashv$

We will now introduce a technical side-condition that ensures that we always have enough witness constants.

DEFINITION 6.3 (Sufficiently Σ-pure). Let Σ be a signature and Φ be a set of Σ-sentences. Φ is called *sufficiently Σ-pure* if for each type α there is a set $\mathcal{P}_\alpha \subseteq \Sigma_\alpha$ of parameters with equal cardinality to $\mathrm{wff}_\alpha(\Sigma)$, such that the elements of $\mathcal{P}_\alpha$ do not occur in the sentences of Φ.

This can be obtained in practice by enriching the signature with spurious parameters. Another way would be to use specially marked variables (which may never be instantiated) as in [36]. Note that for any set to be sufficiently Σ-pure, Σ_α must be infinite for each type α, since we have assumed that $\mathcal{V}_\alpha \subseteq \mathrm{wff}(\Sigma)$ are infinite. Recall that in Remark 3.16 we assumed every Σ_α has a common (infinite) cardinality $\aleph_s$ for every type α. (One could easily show that no set of Σ-sentences could be sufficiently pure if, for example, Σ_ι is countable while $\Sigma_{\iota\to\iota}$ is uncountable. In such a case $\mathrm{wff}_\alpha(\Sigma)$ is uncountable for every type α so one could not satisfy the sufficient purity condition at type ι.)

NOTATION 6.4. For reasons of legibility we will write $S * a$ for $S \cup \{a\}$, where S is a set. We will use this notation with the convention that $*$ associates to the left.

DEFINITION 6.5 (Properties for abstract consistency classes). Let Γ_Σ be a class of sets of Σ-sentences. We define the following properties of Γ_Σ, where $\Phi \in \Gamma_\Sigma$, α, $\beta \in \mathscr{T}$, $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_o$, $\boldsymbol{F} \in \mathrm{cwff}_{\alpha\to o}$, and $\boldsymbol{G}, \boldsymbol{H}, (\lambda X_\alpha \boldsymbol{.} \boldsymbol{M}), (\lambda X_\alpha \boldsymbol{.} \boldsymbol{N}) \in \mathrm{cwff}_{\alpha\to\beta}$ are arbitrary.

∇_c: If $\boldsymbol{A}$ is atomic, then $\boldsymbol{A} \notin \Phi$ or $\neg\boldsymbol{A} \notin \Phi$.
$\nabla_\neg$: If $\neg\neg\boldsymbol{A} \in \Phi$, then $\Phi * \boldsymbol{A} \in \Gamma_\Sigma$.
∇_β: If $\boldsymbol{A} \equiv_\beta \boldsymbol{B}$ and $\boldsymbol{A} \in \Phi$, then $\Phi * \boldsymbol{B} \in \Gamma_\Sigma$.
∇_η: If $\boldsymbol{A} \equiv_{\beta\eta} \boldsymbol{B}$ and $\boldsymbol{A} \in \Phi$, then $\Phi * \boldsymbol{B} \in \Gamma_\Sigma$.
$\nabla_\vee$: If $\boldsymbol{A} \vee \boldsymbol{B} \in \Phi$, then $\Phi * \boldsymbol{A} \in \Gamma_\Sigma$ or $\Phi * \boldsymbol{B} \in \Gamma_\Sigma$.
$\nabla_\wedge$: If $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \Phi$, then $\Phi * \neg\boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$.
$\nabla_\forall$: If $\Pi^\alpha \boldsymbol{F} \in \Phi$, then $\Phi * \boldsymbol{FW} \in \Gamma_\Sigma$ for each $\boldsymbol{W} \in \mathrm{cwff}_\alpha$.
$\nabla_\exists$: If $\neg\Pi^\alpha \boldsymbol{F} \in \Phi$, then $\Phi * \neg(\boldsymbol{F}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
$\nabla_\mathfrak{b}$: If $\neg(\boldsymbol{A} \doteq^o \boldsymbol{B}) \in \Phi$, then $\Phi * \boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$ or $\Phi * \neg\boldsymbol{A} * \boldsymbol{B} \in \Gamma_\Sigma$.
∇_ξ: If $\neg(\lambda X_\alpha \boldsymbol{.} \boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X_\alpha \boldsymbol{.} \boldsymbol{N}) \in \Phi$, then $\Phi * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N}) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
$\nabla_\mathfrak{f}$: If $\neg(\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H}) \in \Phi$, then $\Phi * \neg(\boldsymbol{G}w \doteq^\beta \boldsymbol{H}w) \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in any sentence of Φ.
∇_{sat}: Either $\Phi * \boldsymbol{A} \in \Gamma_\Sigma$ or $\Phi * \neg\boldsymbol{A} \in \Gamma_\Sigma$.

For the optional case of primitive equality, i.e., when $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ for all types α, we now add a set of further properties. While our first choice will be to combine the $\nabla_=^r$ property with $\nabla_=^{\doteq}$, we will later show that other pair combinations from this set are equivalent.

DEFINITION 6.6 (Properties for abstract consistency classes). Suppose $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ for all types α. Let Γ_Σ be a class of sets of Σ-sentences. We define for $\Phi \in \Gamma_\Sigma$, $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_\alpha$ and $\boldsymbol{F} \in \mathrm{cwff}_o$ where $\boldsymbol{F}$ has a subterm of type α at position p:

$\nabla_=^r$: $\neg(\boldsymbol{A} =^\alpha \boldsymbol{A}) \notin \Phi$.
$\nabla_=^s$: If $\boldsymbol{F}[\boldsymbol{A}]_p \in \Phi$ and $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \Phi$, then $\Phi * \boldsymbol{F}[\boldsymbol{B}]_p \in \Gamma_\Sigma$.[9]
$\nabla_=^{\doteq}$: If $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \Phi$, then $\Phi * \boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \Gamma_\Sigma$.
$\nabla_{\doteq}^{=}$: If $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \Phi$, then $\Phi * \boldsymbol{A} =^\alpha \boldsymbol{B} \in \Gamma_\Sigma$.
$\nabla_{=^-}^{\doteq^-}$: If $\neg(\boldsymbol{A} =^\alpha \boldsymbol{B}) \in \Phi$, then $\Phi * \neg(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \Gamma_\Sigma$.
$\nabla_{\doteq^-}^{=^-}$: If $\neg(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \Phi$, then $\Phi * \neg(\boldsymbol{A} =^\alpha \boldsymbol{B}) \in \Gamma_\Sigma$.

DEFINITION 6.7 (Abstract consistency classes). Let Σ be a signature and Γ_Σ be a class of sets of Σ-sentences that is closed under subsets. If $\nabla_c, \nabla_\neg, \nabla_\beta, \nabla_\vee, \nabla_\wedge, \nabla_\forall$ and $\nabla_\exists$ are valid for Γ_Σ, then Γ_Σ is called an *abstract consistency class* for Σ-models. Furthermore, when $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ for all types α and the properties $\nabla_=^r$ and $\nabla_=^{\doteq}$ are valid then Γ_Σ is called an *abstract consistency class with primitive equality*. In the following we often simply use the phrase abstract consistency class to refer to an abstract consistency class with or without primitive equality. We will denote

[9]Although this resembles Lemma 3.25 which required property ξ, it is far weaker since $\boldsymbol{A}$ and $\boldsymbol{B}$ must be closed.

the collection of abstract consistency classes (with primitive equality) by $\mathfrak{Acc}_\beta$. Similarly, we introduce the following collections of specialized abstract consistency classes (with primitive equality): $\mathfrak{Acc}_{\beta\eta}$, $\mathfrak{Acc}_{\beta\xi}$, $\mathfrak{Acc}_{\beta\mathfrak{f}}$, $\mathfrak{Acc}_{\beta\mathfrak{b}}$, $\mathfrak{Acc}_{\beta\eta\mathfrak{b}}$, $\mathfrak{Acc}_{\beta\xi\mathfrak{b}}$, $\mathfrak{Acc}_{\beta\mathfrak{f}\mathfrak{b}}$, where we indicate by indices which additional properties from $\{\nabla_\eta, \nabla_\xi, \nabla_\mathfrak{f}, \nabla_\mathfrak{b}\}$ are required.

Remark 6.8. If primitive equality is not in the signature, $\mathfrak{Acc}_\beta$ corresponds to the abstract consistency property discussed by Andrews in [1]. The only (technical) differences correspond to $\alpha\beta$-conversion. In [1], α-conversion is handled in the ∇_β rule using α-standardized forms. Also, we have defined the ∇_β rule to work with β-conversion instead of β-reduction. We prefer this stronger version of ∇_β over the weaker option "If $\mathbf{A} \in \Phi$, then $\Phi * \mathbf{A}{\downarrow_\beta} \in \Gamma_\Sigma$" since it helps to avoid the use of ∇_{sat} in several proofs below. (Note that ∇_β follows from the weaker option and ∇_{sat}.) Furthermore, in practical applications, e.g., proving completeness of calculi, the stronger property is typically as easy to validate as the weaker one. An analogous argument applies to ∇_η.

Remark 6.9. While the work presented in this article is based on the choice of the primitive logical connectives $\neg$, $\vee$, and Π^α (and possibly primitive equality), a means to generalize the framework over the concrete choice of logical primitives is provided by the uniform notation approach as, for instance, given in [22]. It is clearly possible to achieve such a generalization for our framework as well. This can be done in straightforward manner: $\nabla_\wedge$ becomes an α-property, $\nabla_\vee$ becomes a β-property, $\nabla_\forall$ becomes a γ-property, and $\nabla_\exists$ becomes a δ-property. Thus they will have the following form:

α-case: If $\alpha \in \Phi$, then $\Phi * \alpha_1 * \alpha_2 \in \Gamma_\Sigma$.
β-case: If $\beta \in \Phi$, then $\Phi * \beta_1 \in \Gamma_\Sigma$ or $\Phi * \beta_2 \in \Gamma_\Sigma$.
γ-case: If $\gamma \in \Phi$, then $\Phi * \gamma \mathbf{W} \in \Gamma_\Sigma$ for each $\mathbf{W} \in \mathrm{cwff}_\alpha$.
δ-case: If $\delta \in \Phi$, then $\Phi * \delta w \in \Gamma_\Sigma$ for any parameter $w_\alpha \in \Sigma$ which does not occur in any sentence of Φ.

We often refer to property ∇_c as "atomic consistency". The next lemma shows that we also have the corresponding property for non-atoms.

Lemma 6.10 (Non-atomic consistency). *Let Γ_Σ be an abstract consistency class and $\mathbf{A} \in \mathrm{cwff}_o(\Sigma)$, then for all $\Phi \in \Gamma_\Sigma$ we have $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$.*

Proof following a similar argument in [1], Lemma 3.3.3. If for some $\Phi \in \Gamma_\Sigma$ and $\mathbf{A} \in \mathrm{cwff}_o(\Sigma)$ we have $\mathbf{A} \in \Phi$ and $\neg\mathbf{A} \in \Phi$, then $\{\mathbf{A}, \neg\mathbf{A}\} \in \Gamma_\Sigma$ since Γ_Σ is closed under subsets. Furthermore, using ∇_β and closure under subsets we can assume such an $\mathbf{A}$ is β-normal. We prove $\{\mathbf{A}, \neg\mathbf{A}\} \notin \Gamma_\Sigma$ for any β-normal $\mathbf{A} \in \mathrm{cwff}_o(\Sigma)$ by induction on the number of logical constants in $\mathbf{A}$.

If $\mathbf{A}$ is atomic (which includes primitive equations), this follows immediately from ∇_c. Suppose $\mathbf{A} \equiv \neg\mathbf{B}$ for some $\mathbf{B} \in \mathrm{cwff}_o(\Sigma)$ and $\{\neg\mathbf{B}, \neg\neg\mathbf{B}\} \in \Gamma_\Sigma$. By $\nabla_\neg$ and closure under subsets, we have $\{\neg\mathbf{B}, \mathbf{B}\} \in \Gamma_\Sigma$, contradicting the induction hypothesis for $\mathbf{B}$. Suppose $\mathbf{A} \equiv \mathbf{B} \vee \mathbf{C}$ for some $\mathbf{B}, \mathbf{C} \in \mathrm{cwff}_o(\Sigma)$ and $\{\mathbf{B} \vee \mathbf{C}, \neg(\mathbf{B} \vee \mathbf{C})\} \in \Gamma_\Sigma$. By $\nabla_\vee$, $\nabla_\wedge$ and closure under subsets, we have either $\{\mathbf{B}, \neg\mathbf{B}\} \in \Gamma_\Sigma$ or $\{\mathbf{C}, \neg\mathbf{C}\} \in \Gamma_\Sigma$, contradicting the induction hypotheses for $\mathbf{B}$ and $\mathbf{C}$. Suppose $\mathbf{A} \equiv \Pi^\alpha\mathbf{B}$ for some $\mathbf{B} \in \mathrm{cwff}_{\alpha\to o}(\Sigma)$ and $\{\Pi^\alpha\mathbf{B}, \neg(\Pi^\alpha\mathbf{B})\} \in \Gamma_\Sigma$. Since Σ_α is assumed to be infinite (by Remark 3.16), there is a parameter $w_\alpha \in \Sigma_\alpha$ which does not occur in $\mathbf{A}$. Since

w is a parameter, the sentence $\boldsymbol{B}w$ clearly has one less logical constant than $\Pi^\alpha \boldsymbol{B}$. However, we cannot directly apply the induction hypothesis as $\boldsymbol{B}w$ may not be β-normal. Since $\boldsymbol{B}$ is β-normal, the only way $\boldsymbol{B}w$ can fail to be β-normal is if $\boldsymbol{B}$ has the form $\lambda X_\alpha.\boldsymbol{C}$ for some $\boldsymbol{C} \in \mathrm{wff}_o(\Sigma)$ where $\mathrm{free}(\boldsymbol{C}) \subseteq \{X_\alpha\}$. In this case, it is easy to show that the reduct $[w/X]\boldsymbol{C}$ is β-normal and contains the same number of logical constants as $\boldsymbol{B}$. In either case, we can let $\boldsymbol{N}$ be the β-normal form of $\boldsymbol{B}w$ and apply the induction hypothesis to obtain $\{\boldsymbol{N}, \neg\boldsymbol{N}\} \notin \Gamma_\Sigma$. On the other hand, $\nabla_\exists$, $\nabla_\forall$, ∇_β and closure under subsets implies $\{\boldsymbol{N}, \neg\boldsymbol{N}\} \in \Gamma_\Sigma$, a contradiction. $\dashv$

REMARK 6.11. Note that for the connectives $\vee$ and Π^α there is a positive and a negative condition given in the definition above, namely $\nabla_\vee/\nabla_\wedge$ for $\vee$ and $\nabla_\forall/\nabla_\exists$ for Π^α. For $\doteq^o$ and $\doteq^{\alpha\to\beta}$ the situation is different since we need only conditions for the negative cases. Positive counterparts can be inferred by expanding the Leibniz definition of equality (cf. Lemma 6.12).

LEMMA 6.12 (Leibniz equality). *Let Γ_Σ be an abstract consistency class. The following properties are valid for all $\Phi \in \Gamma_\Sigma$, $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_o(\Sigma)$, $\boldsymbol{C} \in \mathrm{cwff}_\alpha(\Sigma)$ and $\boldsymbol{F}, \boldsymbol{G} \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma)$.*

$\nabla^r_{\doteq}$: $\neg(\boldsymbol{C} \doteq^\alpha \boldsymbol{C}) \notin \Phi$.

$\nabla^\to_{\doteq}$: *If $\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G} \in \Phi$, then $\Phi * \boldsymbol{FW} \doteq^\beta \boldsymbol{GW} \in \Gamma_\Sigma$ for any closed $\boldsymbol{W} \in \mathrm{cwff}_\alpha(\Sigma)$.*

$\nabla^o_{\doteq}$: *If $\boldsymbol{A} \doteq^o \boldsymbol{B} \in \Phi$, then $\Phi * \boldsymbol{A} * \boldsymbol{B} \in \Gamma_\Sigma$ or $\Phi * \neg\boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$.*

PROOF. To show $\nabla^r_{\doteq}$, assume $\neg(\boldsymbol{C} \doteq \boldsymbol{C}) \in \Phi$. By subset closure $\{\neg(\boldsymbol{C} \doteq \boldsymbol{C})\} \in \Gamma_\Sigma$ and by $\nabla_\exists$ with some parameter p which does not occur in $\boldsymbol{C}$ and ∇_β we get $\{\neg(\boldsymbol{C} \doteq \boldsymbol{C}), \neg(\neg p\boldsymbol{C} \vee p\boldsymbol{C})\} \in \Gamma_\Sigma$. The contradiction follows by $\nabla_\wedge$, $\nabla_\neg$ and ∇_c. So, $\nabla^r_{\doteq}$ holds.

To show $\nabla^\to_{\doteq}$, suppose $\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G} \in \Phi$. By application of $\nabla_\forall$ with $\lambda X_{\alpha\to\beta}.\boldsymbol{FW} \doteq X\boldsymbol{W}$ and ∇_β we have $\Phi * (\neg(\boldsymbol{FW} \doteq \boldsymbol{FW}) \vee \boldsymbol{FW} \doteq \boldsymbol{GW}) \in \Gamma_\Sigma$. By $\nabla_\vee$ and subset closure we get $\Phi * \neg(\boldsymbol{FW} \doteq \boldsymbol{FW}) \in \Gamma_\Sigma$ or $\Phi * \boldsymbol{FW} \doteq \boldsymbol{GW} \in \Gamma_\Sigma$. The latter proves the assertion since the first option is ruled out by $\nabla^r_{\doteq}$ (shown above).

To show $\nabla^o_{\doteq}$, suppose $\boldsymbol{A} \doteq^o \boldsymbol{B} \in \Phi$. Applying $\nabla_\forall$ with $\lambda Y.Y$ we have $\Phi * (\lambda P_{o\to o}.\neg P\boldsymbol{A} \vee P\boldsymbol{B})(\lambda Y.Y) \in \Gamma_\Sigma$. By ∇_β and subset closure we get $\Phi * \neg\boldsymbol{A} \vee \boldsymbol{B} \in \Gamma_\Sigma$. Similarly, we further derive by $\nabla_\forall$ with $\lambda Y.\neg Y$, ∇_β, and subset closure that $\Phi * \neg\boldsymbol{A} \vee \boldsymbol{B} * \neg\neg\boldsymbol{A} \vee \neg\boldsymbol{B} \in \Gamma_\Sigma$. By applying $\nabla_\vee$ twice and subset closure we get the following four options: (i) $\Phi * \neg\boldsymbol{A} * \neg\neg\boldsymbol{A} \in \Gamma_\Sigma$, (ii) $\Phi * \neg\boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$, (iii) $\Phi * \boldsymbol{B} * \neg\neg\boldsymbol{A} \in \Gamma_\Sigma$, or (iv) $\Phi * \boldsymbol{B} * \neg\boldsymbol{B} \in \Gamma_\Sigma$. Cases (i) and (iv) are ruled out by non-atomic consistency. In case (iii) we furthermore get by $\nabla_\neg$ and subset closure that $\Phi * \boldsymbol{B} * \boldsymbol{A} \in \Gamma_\Sigma$. Thus, $\Phi * \neg\boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$ or $\Phi * \boldsymbol{B} * \boldsymbol{A} \in \Gamma_\Sigma$. $\dashv$

We could easily add respective properties for symmetry, transitivity, and congruence to the previous lemma. They can be shown analogously, i.e., they also follow from the properties of Leibniz equality.

In contrast to [1], we work with saturated abstract consistency classes in order to simplify the proofs of the model existence theorems. For a discussion of the consequences of this decision, see Section 8.2.

DEFINITION 6.13 (Saturatedness). We call an abstract consistency class Γ_Σ saturated if it satisfies ∇_{sat}.

REMARK 6.14. Clearly, not all abstract consistency classes are saturated, since the empty set is one that is not ($\mathrm{cwff}_o(\Sigma)$ is certainly non-empty since $\forall P_o\text{.}P \in \mathrm{cwff}_o(\Sigma)$).

REMARK 6.15. The saturation condition ∇_{sat} can be very difficult to verify in practice. For example, showing that an abstract consistency class induced from a sequent calculus (as in [1]) is saturated corresponds to showing cut-elimination (cf. [12]). Since Andrews [1] did not use saturation, he could use his results to give a model-theoretic proof of cut-elimination for a sequent calculus. We cannot use the results of this article to obtain similar cut-elimination results.

We now investigate derived properties of primitive equality.

LEMMA 6.16 (Primitive equality). *Let Γ_Σ be an abstract consistency class with primitive equality, i.e., $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ for all types $\alpha \in \mathscr{T}$, where $\nabla^r_{\underline{=}}$ and $\nabla^{\doteq}_{\underline{=}}$ hold. Then $\nabla^{=}_{\underline{\doteq}}$ and $\nabla^s_{\underline{=}}$ are valid. Furthermore, $\nabla^{\doteq^-}_{\underline{=}^-}$ and $\nabla^{=^-}_{\underline{\doteq}^-}$ are valid if Γ_Σ is saturated.*

PROOF. To show $\nabla^{=}_{\underline{\doteq}}$ we derive from $(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \Phi$ by $\nabla_\forall$ with $\lambda X_\alpha\text{.}\boldsymbol{A} =^\alpha X$, ∇_β, and subset closure that $\Phi * \neg(\boldsymbol{A} = \boldsymbol{A}) \vee \boldsymbol{A} = \boldsymbol{B} \in \Gamma_\Sigma$. By $\nabla_\vee$ and subset closure we get $\Phi * \neg(\boldsymbol{A} = \boldsymbol{A}) \in \Gamma_\Sigma$ or $\Phi * \boldsymbol{A} = \boldsymbol{B} \in \Gamma_\Sigma$. The assertion follows from the latter option since the former is ruled out by $\nabla^r_{\underline{=}}$.

In order to show $\nabla^s_{\underline{=}}$ let $\boldsymbol{F}[\boldsymbol{A}]_p \in \Phi$, we derive from $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \Phi$ by $\nabla^{\doteq}_{\underline{=}}$ that $\Phi * (\boldsymbol{A} \doteq \boldsymbol{B}) \in \Gamma_\Sigma$. By $\nabla_\forall$ with $\lambda X\text{.}\boldsymbol{F}[X]_p$ (where $X \in \mathscr{V}_\alpha$ does not occur bound in $\boldsymbol{F}[\boldsymbol{A}]_p$), ∇_β, and subset closure we furthermore get that $\Phi * (\neg\boldsymbol{F}[\boldsymbol{A}]_p \vee \boldsymbol{F}[\boldsymbol{B}]_p) \in \Gamma_\Sigma$. Application of $\nabla_\vee$ and subset closure gives us $\Phi * \neg\boldsymbol{F}[\boldsymbol{A}]_p \in \Gamma_\Sigma$ or $\Phi * \boldsymbol{F}[\boldsymbol{B}]_p \in \Gamma_\Sigma$. The assertion follows from the latter option since the former is ruled out by $\boldsymbol{F}[\boldsymbol{A}]_p \in \Phi$ and non-atomic consistency.

The straightforward proof for $\nabla^{=^-}_{\underline{\doteq}^-}$ employs saturation, $\nabla^{\doteq}_{\underline{=}}$, and non-atomic consistency. Similarly, the proof for $\nabla^{\doteq^-}_{\underline{=}^-}$ employs saturation, $\nabla^{=}_{\underline{\doteq}}$, and atomic consistency. ⊣

The next theorem provides some alternatives to our choice of $\nabla^{\doteq}_{\underline{=}}$ and $\nabla^r_{\underline{=}}$ in the definition of abstract consistency classes with primitive equality provided that saturation holds. In practical applications the user may therefore choose the combination that suits best.

THEOREM 6.17 (Alternative properties for primitive equality). *Let Γ_Σ be an abstract consistency class and let $=^\alpha \in \Sigma_{\alpha\to\alpha\to o}$ for all types $\alpha \in \mathscr{T}$. If Γ_Σ is saturated and validates one of the following combinations of properties, then it also validates $\nabla^{\doteq}_{\underline{=}}$ and $\nabla^r_{\underline{=}}$. The combinations are*:

(1) $\nabla^s_{\underline{=}}$ *and* $\nabla^r_{\underline{=}}$.
(2) $\nabla^{\doteq}_{\underline{=}}$ *and* $\nabla^{=}_{\underline{\doteq}}$.
(3) $\nabla^{\doteq^-}_{\underline{=}^-}$ *and* $\nabla^{=^-}_{\underline{\doteq}^-}$.

PROOF. To prove (1) we only have to show $\nabla^{\doteq}_{\underline{=}}$. Let $(\boldsymbol{A} = \boldsymbol{B}) \in \Phi$ and suppose $\Phi * (\boldsymbol{A} \doteq \boldsymbol{B}) \notin \Gamma_\Sigma$. Then by saturation $\Phi * \neg(\boldsymbol{A} \doteq \boldsymbol{B}) \in \Gamma_\Sigma$ and by application of $\nabla^s_{\underline{=}}$ we get a contradiction to $\nabla^r_{\underline{=}}$ (cf. Lemma 6.12).

To prove (2) we only have to show $\nabla^r_{\underline{=}}$. Since $\Phi * \neg(\boldsymbol{A} \doteq \boldsymbol{A}) \notin \Gamma_\Sigma$ by $\nabla^r_{\underline{=}}$ we get by saturation $\Phi * \boldsymbol{A} \doteq \boldsymbol{A} \in \Gamma_\Sigma$. By $\nabla^{=}_{\underline{\doteq}}$ and subset closure, we have $\Phi * \boldsymbol{A} = \boldsymbol{A} \in \Gamma_\Sigma$. By atomic consistency, we have $\neg(\boldsymbol{A} = \boldsymbol{A}) \notin \Phi$.

For (3) we first show $\nabla^r_{=}$. Suppose $\neg(\boldsymbol{A} = \boldsymbol{A}) \in \Phi$. Then by $\nabla^{\doteq^-}_{=^-}$ we get $\Phi * \neg(\boldsymbol{A} \doteq \boldsymbol{A}) \in \Gamma_\Sigma$ contradicting $\nabla^r_{\doteq}$. To show $\nabla^{\doteq}_{=}$ let $\boldsymbol{A} = \boldsymbol{B} \in \Phi$ and suppose $\Phi * \boldsymbol{A} \doteq \boldsymbol{B} \notin \Gamma_\Sigma$. By saturation we get $\Phi * \neg(\boldsymbol{A} \doteq \boldsymbol{B}) \in \Gamma_\Sigma$ and by application of $\nabla^{=^-}_{\doteq^-}$ we get a contradiction to atomic consistency. ⊣

Lemma 6.18 (Compactness of abstract consistency classes). *For each abstract consistency class Γ_Σ there exists a compact abstract consistency class Γ'_Σ satisfying the same ∇_* properties such that $\Gamma_\Sigma \subseteq \Gamma'_\Sigma$.*

Proof (following and extending [6], Proposition 2506). We choose $\Gamma'_\Sigma := \{\, \Phi \subseteq \mathrm{cwff}_o \mid \text{every finite subset of } \Phi \text{ is in } \Gamma_\Sigma \,\}$. Now suppose that $\Phi \in \Gamma_\Sigma$. Γ_Σ is closed under subsets, so every finite subset of Φ is in Γ_Σ and thus $\Phi \in \Gamma'_\Sigma$. Hence $\Gamma_\Sigma \subseteq \Gamma'_\Sigma$.

Next let us show that Γ'_Σ is compact. Suppose $\Phi \in \Gamma'_\Sigma$ and Ψ is an arbitrary finite subset of Φ. By definition of Γ'_Σ all finite subsets of Φ are in Γ_Σ and therefore $\Psi \in \Gamma'_\Sigma$. Thus all finite subsets of Φ are in Γ'_Σ whenever Φ is in Γ'_Σ. On the other hand, suppose all finite subsets of Φ are in Γ'_Σ. Then by the definition of Γ'_Σ the finite subsets of Φ are also in Γ_Σ, so $\Phi \in \Gamma'_\Sigma$. Thus Γ'_Σ is compact. Note that by Lemma 6.2 we have that Γ'_Σ is closed under subsets.

Next we show that if Γ_Σ satisfies ∇_*, then Γ'_Σ satisfies ∇_*.

∇_c: Let $\Phi \in \Gamma'_\Sigma$ and suppose there is an atom $\boldsymbol{A}$, such that $\{\boldsymbol{A}, \neg\boldsymbol{A}\} \subseteq \Phi$. $\{\boldsymbol{A}, \neg\boldsymbol{A}\}$ is clearly a finite subset of Φ and hence $\{\boldsymbol{A}, \neg\boldsymbol{A}\} \in \Gamma_\Sigma$ contradicting ∇_c for Γ_Σ.

$\nabla_\neg$: Let $\Phi \in \Gamma'_\Sigma$, $\neg\neg\boldsymbol{A} \in \Phi$, Ψ be any finite subset of $\Phi * \boldsymbol{A}$, and $\Theta := (\Psi \setminus \{\boldsymbol{A}\}) * \neg\neg\boldsymbol{A}$. Θ is a finite subset of Φ, so $\Theta \in \Gamma_\Sigma$. Since Γ_Σ is an abstract consistency class and $\neg\neg\boldsymbol{A} \in \Theta$, we get $\Theta * \boldsymbol{A} \in \Gamma_\Sigma$ by $\nabla_\neg$ for Γ_Σ. We know that $\Psi \subseteq \Theta * \boldsymbol{A}$ and Γ_Σ is closed under subsets, so $\Psi \in \Gamma_\Sigma$. Thus every finite subset Ψ of $\Phi * \boldsymbol{A}$ is in Γ_Σ and therefore by definition $\Phi * \boldsymbol{A} \in \Gamma'_\Sigma$.

$\nabla_\beta, \nabla_\eta, \nabla_\vee, \nabla_\wedge, \nabla_\forall, \nabla_\exists$: Analogous to $\nabla_\neg$.

∇_ξ: Let $\Phi \in \Gamma'_\Sigma$, $\neg(\lambda X_\alpha.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \Phi$ and Ψ be any finite subset of $\Phi * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})$, where $w \in \Sigma_\alpha$ is a parameter that does not occur in any sentence of Φ. We show that $\Psi \in \Gamma_\Sigma$. Clearly $\Theta := (\Psi \setminus \{\neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})\}) * \neg(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N})$ is a finite subset of Φ and therefore $\Theta \in \Gamma_\Sigma$. Since Γ_Σ satisfies ∇_ξ and $\neg(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \Theta$, we have $\Theta * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N}) \in \Gamma_\Sigma$. Furthermore, $\Psi \subseteq \Theta * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})$ and Γ_Σ is closed under subsets, so $\Psi \in \Gamma_\Sigma$. Thus every finite subset Ψ of $\Phi * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})$ is in Γ_Σ, and therefore by definition we have $\Phi * \neg([w/X]\boldsymbol{M} \doteq^\alpha [w/X]\boldsymbol{N}) \in \Gamma'_\Sigma$.

$\nabla_{\mathfrak{f}}$: Analogous to ∇_ξ.

$\nabla_{\mathfrak{b}}$: Let $\Phi \in \Gamma'_\Sigma$ with $\neg(\boldsymbol{A} \doteq \boldsymbol{B}) \in \Phi$. Assume $\Phi * \boldsymbol{A} * \neg\boldsymbol{B} \notin \Gamma_\Sigma$ and $\Phi * \neg\boldsymbol{A} * \boldsymbol{B} \notin \Gamma_\Sigma$. Then there exists finite subsets Φ_1 and Φ_2 of Φ, such that $\Phi_1 * \boldsymbol{A} * \neg\boldsymbol{B} \notin \Gamma_\Sigma$ and $\Phi_2 * \neg\boldsymbol{A} * \boldsymbol{B} \notin \Gamma_\Sigma$. Now we choose $\Phi_3 := \Phi_1 \cup \Phi_2 * \neg(\boldsymbol{A} \doteq \boldsymbol{B})$. Obviously Φ_3 is a finite subset of Φ and therefore $\Phi_3 \in \Gamma_\Sigma$. Since Γ_Σ satisfies $\nabla_{\mathfrak{b}}$, we have that $\Phi_3 * \boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$ or $\Phi_3 * \neg\boldsymbol{A} * \boldsymbol{B} \in \Gamma_\Sigma$. From this and the fact that Γ_Σ is closed under subsets we get that $\Phi_1 * \boldsymbol{A} * \neg\boldsymbol{B} \in \Gamma_\Sigma$ or $\Phi_2 * \neg\boldsymbol{A} * \boldsymbol{B} \in \Gamma_\Sigma$, which contradicts our assumption.

∇_{sat}: Let $\Phi \in \Gamma'_\Sigma$. Assume neither $\Phi * \boldsymbol{A}$ nor $\Phi * \neg\boldsymbol{A}$ is in Γ'_Σ. Then there are finite subsets Φ_1 and Φ_2 of Φ, such that $\Phi_1 * \boldsymbol{A} \notin \Gamma_\Sigma$ and $\Phi_2 * \neg\boldsymbol{A} \notin \Gamma_\Sigma$. As $\Psi := \Phi_1 \cup \Phi_2$ is a finite subset of Φ, we have $\Psi \in \Gamma_\Sigma$. Furthermore,

$\Psi * \boldsymbol{A} \in \Gamma_\Sigma$ or $\Psi * \neg\boldsymbol{A} \in \Gamma_\Sigma$ because Γ_Σ is saturated. Γ_Σ is closed under subsets, so $\Phi_1 * \boldsymbol{A} \in \Gamma_\Sigma$ or $\Phi_2 * \neg\boldsymbol{A} \in \Gamma_\Sigma$. This is a contradiction, so we can conclude that if $\Phi \in \Gamma'_\Sigma$, then $\Phi * \boldsymbol{A} \in \Gamma'_\Sigma$ or $\Phi * \neg\boldsymbol{A} \in \Gamma'_\Sigma$.

In case primitive equality is present in the signature, we check the corresponding properties.

$\nabla^r_=$: Let $\Phi \in \Gamma'_\Sigma$ and assume $\neg(\boldsymbol{A} =^\alpha \boldsymbol{A}) \in \Phi$. $\{\neg(\boldsymbol{A} =^\alpha \boldsymbol{A})\}$ is clearly a finite subset of Φ and hence $\{\neg(\boldsymbol{A} =^\alpha \boldsymbol{A})\} \in \Gamma_\Sigma$ contradicting $\nabla^r_=$ in Γ_Σ.

$\nabla^{\doteq}_=, \nabla^s_=, \nabla^{=}_{\doteq}, \nabla^{\doteq^-}_{=^-}, \nabla^{=^-}_{\doteq^-}$ Analogous to $\nabla_\neg$. ⊣

6.2. Hintikka sets. Hintikka sets connect syntax with semantics as they provide the basis for the model constructions in the model existence theorems. We have defined eight different notions of abstract consistency classes by first defining properties ∇_*, then specifying which should hold in $\mathfrak{Acc}_*$. Similarly, we define Hintikka sets by first defining the desired properties.

Definition 6.19 (Σ-Hintikka properties). Let $\mathcal{H}$ be a set of sentences. We define the following properties which $\mathcal{H}$ may satisfy, where $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_o$, $\boldsymbol{C}, \boldsymbol{D} \in \mathrm{cwff}_\alpha$, $\boldsymbol{F} \in \mathrm{cwff}_{\alpha\to o}$, and $(\lambda X_\alpha.\boldsymbol{M}), (\lambda X.\boldsymbol{N}), \boldsymbol{G}, \boldsymbol{H} \in \mathrm{cwff}_{\alpha\to\beta}$:

$\vec{\nabla}_c$: $\boldsymbol{A} \notin \mathcal{H}$ or $\neg\boldsymbol{A} \notin \mathcal{H}$.
$\vec{\nabla}_\neg$: If $\neg\neg\boldsymbol{A} \in \mathcal{H}$, then $\boldsymbol{A} \in \mathcal{H}$.
$\vec{\nabla}_\beta$: If $\boldsymbol{A} \in \mathcal{H}$ and $\boldsymbol{A} \equiv_\beta \boldsymbol{B}$, then $\boldsymbol{B} \in \mathcal{H}$.
$\vec{\nabla}_\eta$: If $\boldsymbol{A} \in \mathcal{H}$ and $\boldsymbol{A} \equiv_{\beta\eta} \boldsymbol{B}$, then $\boldsymbol{B} \in \mathcal{H}$.
$\vec{\nabla}_\vee$: If $\boldsymbol{A} \vee \boldsymbol{B} \in \mathcal{H}$, then $\boldsymbol{A} \in \mathcal{H}$ or $\boldsymbol{B} \in \mathcal{H}$.
$\vec{\nabla}_\wedge$: If $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \mathcal{H}$, then $\neg\boldsymbol{A} \in \mathcal{H}$ and $\neg\boldsymbol{B} \in \mathcal{H}$.
$\vec{\nabla}_\forall$: If $\Pi^\alpha \boldsymbol{F} \in \mathcal{H}$, then $\boldsymbol{F}\boldsymbol{W} \in \mathcal{H}$ for each $\boldsymbol{W} \in \mathrm{cwff}_\alpha$.
$\vec{\nabla}_\exists$: If $\neg\Pi^\alpha \boldsymbol{F} \in \mathcal{H}$, then there is a parameter $w_\alpha \in \Sigma_\alpha$ such that $\neg(\boldsymbol{F}w) \in \mathcal{H}$.
$\vec{\nabla}_\mathfrak{b}$: If $\neg(\boldsymbol{A} \doteq^o \boldsymbol{B}) \in \mathcal{H}$, then $\{\boldsymbol{A}, \neg\boldsymbol{B}\} \subseteq \mathcal{H}$ or $\{\neg\boldsymbol{A}, \boldsymbol{B}\} \subseteq \mathcal{H}$.
$\vec{\nabla}_\xi$: If $\neg(\lambda X_\alpha.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \mathcal{H}$, then there is a parameter $w_\alpha \in \Sigma_\alpha$ such that $\neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N}) \in \mathcal{H}$.
$\vec{\nabla}_\mathfrak{f}$: If $\neg(\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H}) \in \mathcal{H}$, then there is a parameter $w_\alpha \in \Sigma_\alpha$ such that $\neg(\boldsymbol{G}w \doteq^\beta \boldsymbol{H}w) \in \mathcal{H}$.
$\vec{\nabla}_{sat}$: Either $\boldsymbol{A} \in \mathcal{H}$ or $\neg\boldsymbol{A} \in \mathcal{H}$.
$\vec{\nabla}^r_=$: $\neg(\boldsymbol{C} =^\alpha \boldsymbol{C}) \notin \mathcal{H}$.
$\vec{\nabla}^{\doteq}_=$: If $\boldsymbol{C} =^\alpha \boldsymbol{D} \in \mathcal{H}$, then $\boldsymbol{C} \doteq^\alpha \boldsymbol{D} \in \mathcal{H}$.

Definition 6.20 (Σ-Hintikka set). A set $\mathcal{H}$ of sentences is called a Σ-*Hintikka set* if it satisfies $\vec{\nabla}_c$, $\vec{\nabla}_\neg$, $\vec{\nabla}_\beta$, $\vec{\nabla}_\vee$, $\vec{\nabla}_\wedge$, $\vec{\nabla}_\forall$ and $\vec{\nabla}_\exists$. When primitive equality is present in the signature and $\mathcal{H}$ is a Hintikka set satisfying $\vec{\nabla}^r_=$ and $\vec{\nabla}^{\doteq}_=$ we call $\mathcal{H}$ a Σ-*Hintikka set with primitive equality*. We define the following collections of Hintikka sets (with primitive equality): $\mathfrak{Hint}_\beta$, $\mathfrak{Hint}_{\beta\eta}$, $\mathfrak{Hint}_{\beta\xi}$, $\mathfrak{Hint}_{\beta\mathfrak{f}}$, $\mathfrak{Hint}_{\beta\mathfrak{b}}$, $\mathfrak{Hint}_{\beta\eta\mathfrak{b}}$, $\mathfrak{Hint}_{\beta\xi\mathfrak{b}}$, and $\mathfrak{Hint}_{\beta\mathfrak{f}\mathfrak{b}}$, where we indicate by indices which additional properties from $\{\vec{\nabla}_\eta, \vec{\nabla}_\xi, \vec{\nabla}_\mathfrak{f}, \vec{\nabla}_\mathfrak{b}\}$ are required. If primitive equality is in the signature, we require $\mathcal{H} \in \mathfrak{Hint}_*$ to be a Hintikka set with primitive equality.

We will construct Hintikka sets as maximal elements of abstract consistency classes. To obtain a Hintikka set, we must explicitly show the property $\vec{\nabla}_\exists$ (and $\vec{\nabla}_\xi$

or $\vec{\nabla}_{\mathfrak{f}}$ when appropriate). This will ensure that Hintikka sets have enough parameters which act as witnesses.

LEMMA 6.21 (Hintikka lemma). *Let* Γ_Σ *be an abstract consistency class in* $\mathfrak{Acc}_*$. *Suppose a set* $\mathscr{H} \in \Gamma_\Sigma$ *satisfies the following properties*:

(1) $\mathscr{H}$ *is subset-maximal in* Γ_Σ (*i.e., for each sentence* $\boldsymbol{D} \in \mathrm{cwff}_o$ *such that* $\mathscr{H} * \boldsymbol{D} \in \Gamma_\Sigma$, *we already have* $\boldsymbol{D} \in \mathscr{H}$).
(2) $\mathscr{H}$ *satisfies* $\vec{\nabla}_\exists$.
(3) *If* $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$, *then* $\vec{\nabla}_\xi$ *holds in* $\mathscr{H}$.
(4) *If* $* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$, *then* $\vec{\nabla}_{\mathfrak{f}}$ *holds in* $\mathscr{H}$.

Then, $\mathscr{H} \in \mathfrak{Hint}_*$. *Furthermore, if* Γ_Σ *is saturated, then* $\mathscr{H}$ *satisfies* $\vec{\nabla}_{sat}$.

PROOF. $\mathscr{H}$ satisfies $\vec{\nabla}_\exists$ by assumption. Also, if $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$ ($* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$), then we have explicitly assumed $\mathscr{H}$ satisfies $\vec{\nabla}_\xi$ ($\vec{\nabla}_{\mathfrak{f}}$). The fact that $\mathscr{H} \in \Gamma_\Sigma$ satisfies $\vec{\nabla}_c$ follows directly from non-atomic consistency (Lemma 6.10). Similarly, if primitive equality is in the signature, then $\mathscr{H}$ satisfies $\vec{\nabla}^r_=$ since $\mathscr{H} \in \Gamma_\Sigma$ and Γ_Σ satisfies $\nabla^r_=$. Every other $\vec{\nabla}_*$ property follows directly from the corresponding ∇_* property and maximality of $\mathscr{H}$ in Γ_Σ. For example, to show $\vec{\nabla}_\neg$, suppose $\neg\neg\boldsymbol{A} \in \mathscr{H}$. By $\nabla_\neg$, we know $\mathscr{H} * \boldsymbol{A} \in \Gamma_\Sigma$. By maximality of $\mathscr{H}$, we have $\boldsymbol{A} \in \mathscr{H}$. Checking $\vec{\nabla}_\beta$, $\vec{\nabla}_\eta$ (if $* \in \{\beta\eta, \beta\eta\mathfrak{b}\}$), $\vec{\nabla}_\wedge$, $\vec{\nabla}_\forall$, and $\vec{\nabla}^{\doteq}_=$ hold for $\mathscr{H}$ follows exactly this same pattern. Checking $\vec{\nabla}_\vee$, $\vec{\nabla}_{\mathfrak{b}}$ (if $* \in \{\beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$) and $\vec{\nabla}_{sat}$ (if Γ_Σ is saturated) follows a similar pattern, but with a simple case analysis. For example, to check $\vec{\nabla}_{sat}$, given $\boldsymbol{A} \in \mathrm{cwff}_o(\Sigma)$, ∇_{sat} implies $\mathscr{H} * \boldsymbol{A} \in \Gamma_\Sigma$ or $\mathscr{H} * \neg\boldsymbol{A} \in \Gamma_\Sigma$. So, either $\boldsymbol{A} \in \mathscr{H}$ or $\neg\boldsymbol{A} \in \mathscr{H}$. ⊣

It is worth noting that the converse of $\vec{\nabla}^{\doteq}_=$ also holds in Hintikka sets with primitive equality.

LEMMA 6.22. *Suppose primitive equality is in the signature and* $\mathscr{H}$ *is a Hintikka set with primitive equality. Then, we have the following property for every type* α *and* $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_\alpha(\Sigma)$:

$\overline{\nabla}^{\doteq}_=$: $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \mathscr{H}$ *iff* $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \mathscr{H}$.

PROOF. If $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \mathscr{H}$, then $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \mathscr{H}$ by $\vec{\nabla}^{\doteq}_=$. For the converse direction assume that $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \mathscr{H}$. From this we get by $\vec{\nabla}_\forall$ with $\lambda X.\boldsymbol{A} = X$ and ∇_β that $\neg(\boldsymbol{A} = \boldsymbol{A}) \vee \boldsymbol{A} = \boldsymbol{B} \in \mathscr{H}$. Since $\neg(\boldsymbol{A} = \boldsymbol{A}) \notin \mathscr{H}$ by $\vec{\nabla}^r_=$, $\vec{\nabla}_\vee$ implies $\boldsymbol{A} =^\alpha \boldsymbol{B} \in \mathscr{H}$. ⊣

It is helpful to note the following properties of Leibniz equality in Hintikka sets.

LEMMA 6.23. *Suppose* $\mathscr{H}$ *is a Hintikka set. For any* $\boldsymbol{F}, \boldsymbol{G} \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma)$ *and* $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathrm{cwff}_\alpha(\Sigma)$ (*for types* α *and* β), *we have the following*:

$\vec{\nabla}^r_{\doteq}$: $\neg(\boldsymbol{A} \doteq^\alpha \boldsymbol{A}) \notin \mathscr{H}$.

$\vec{\nabla}^{tr}_{\doteq}$: *If* $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \mathscr{H}$ *and* $\boldsymbol{B} \doteq^\alpha \boldsymbol{C} \in \mathscr{H}$, *then* $\boldsymbol{A} \doteq^\alpha \boldsymbol{C} \in \mathscr{H}$.

$\vec{\nabla}^{\to}_{\doteq}$: *If* $(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G}) \in \mathscr{H}$ *and* $(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \mathscr{H}$, *then* $(\boldsymbol{FA} \doteq^\beta \boldsymbol{GB}) \in \mathscr{H}$.

PROOF. To show $\vec{\nabla}^r_{\doteq}$, suppose $\neg(\boldsymbol{A} \doteq^\alpha \boldsymbol{A}) \in \mathscr{H}$. By $\vec{\nabla}_\exists$ and $\vec{\nabla}_\beta$, there must be some parameter $q_{\alpha\to o}$ such that $\neg(\neg q\boldsymbol{A} \vee q\boldsymbol{A}) \in \mathscr{H}$. By $\vec{\nabla}_\wedge$, we have $\neg\neg q\boldsymbol{A} \in \mathscr{H}$ and $\neg q\boldsymbol{A} \in \mathscr{H}$, contradicting $\vec{\nabla}_c$.

To show $\vec{\nabla}^{tr}_{\doteq}$, suppose $\boldsymbol{A} \doteq^\alpha \boldsymbol{B} \in \mathscr{H}$ and $\boldsymbol{B} \doteq^\alpha \boldsymbol{C} \in \mathscr{H}$. Let $\boldsymbol{Q}_{\alpha\to o}$ be the closed formula $(\lambda X_\alpha.\boldsymbol{A} \doteq^\alpha X)$. Applying $\vec{\nabla}_\forall$ to $\boldsymbol{B} \doteq^\alpha \boldsymbol{C} \in \mathscr{H}$ and $\boldsymbol{Q}$, we know

$\neg(\boldsymbol{QB}) \vee \boldsymbol{QC} \in \mathcal{H}$. By $\vec{\nabla}_\vee$, we know $\neg(\boldsymbol{QB}) \in \mathcal{H}$ or $\boldsymbol{QC} \in \mathcal{H}$. If $\neg(\boldsymbol{QB}) \in \mathcal{H}$, then $\neg(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \mathcal{H}$ by $\vec{\nabla}_\beta$, contradicting $\vec{\nabla}_c$. So, $\boldsymbol{QC} \in \mathcal{H}$ and hence $\boldsymbol{A} \doteq^\alpha \boldsymbol{C} \in \mathcal{H}$ as desired.

To show $\vec{\nabla}_{\doteq}^{\rightarrow}$, let $\boldsymbol{P}_{(\alpha\to\beta)\to o}$ be the closed formula $(\lambda H_{\alpha\to\beta}.\boldsymbol{FA} \doteq^\beta H\boldsymbol{A})$, Applying $\vec{\nabla}_\forall$ to $(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G}) \in \mathcal{H}$ and $\boldsymbol{P}$, we have $\neg(\boldsymbol{PF}) \vee \boldsymbol{PG} \in \mathcal{H}$. By $\vec{\nabla}_\vee$, we know $\neg(\boldsymbol{PF}) \in \mathcal{H}$ or $\boldsymbol{PG} \in \mathcal{H}$. If $\neg(\boldsymbol{PF}) \in \mathcal{H}$, then $\neg(\boldsymbol{FA} \doteq^\beta \boldsymbol{FA}) \in \mathcal{H}$ by $\vec{\nabla}_\beta$, which contradicts $\vec{\nabla}_{\doteq}^r$. So, we must have $\boldsymbol{PG} \in \mathcal{H}$ and hence $(\boldsymbol{FA} \doteq^\beta \boldsymbol{GA}) \in \mathcal{H}$. Let $\boldsymbol{Q}_{\alpha\to o}$ be the closed formula $(\lambda X_\alpha.\boldsymbol{FA} \doteq^\beta \boldsymbol{G}X)$. Applying $\vec{\nabla}_\forall$ and $\vec{\nabla}_\vee$ to $(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \mathcal{H}$, we know $\neg(\boldsymbol{QA}) \in \mathcal{H}$ or $\boldsymbol{QB} \in \mathcal{H}$. If $\neg(\boldsymbol{QA}) \in \mathcal{H}$, then $\neg(\boldsymbol{FA} \doteq^\beta \boldsymbol{GA}) \in \mathcal{H}$ by $\vec{\nabla}_\beta$, contradicting $\vec{\nabla}_c$. So, $\boldsymbol{QB} \in \mathcal{H}$ and hence $(\boldsymbol{FA} \doteq^\beta \boldsymbol{GB}) \in \mathcal{H}$ as desired. ⊣

Whenever a Hintikka set satisfies $\vec{\nabla}_{sat}$, we can prove far more closure properties. For example, we can prove converses of $\vec{\nabla}_\neg$, $\vec{\nabla}_\beta$, $\vec{\nabla}_\vee$, $\vec{\nabla}_\wedge$, $\vec{\nabla}_\forall$, $\vec{\nabla}_\exists$ and $\vec{\nabla}_{=}^{\doteq}$ (when primitive equality is in the signature). Also, if any of $\vec{\nabla}_\eta$, $\vec{\nabla}_\mathfrak{b}$, $\vec{\nabla}_\xi$ or $\vec{\nabla}_\mathfrak{f}$ hold, we can prove the corresponding converse. (We could call these properties $\overleftarrow{\nabla}_*$.) The proofs of the stronger properties $\overline{\nabla}_\neg$ and $\overline{\nabla}_\vee$ in Lemma 6.25 indicate how one would prove any of these converse properties.

Definition 6.24 (Saturated set). We say a set of sentences $\mathcal{H}$ is *saturated* if it satisfies $\vec{\nabla}_{sat}$.

By Lemma 6.21, any Hintikka set constructed as a maximal member of a saturated abstract consistency class will be saturated. However, it is also possible for a maximal member of an abstract consistency class Γ_Σ to be saturated without Γ_Σ being saturated.

Lemma 6.25 (Saturated sets lemma). *Suppose $\mathcal{H}$ is a saturated Hintikka set. Then we have the following properties for every $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_o(\Sigma)$, $\boldsymbol{F} \in \mathrm{cwff}_{\alpha\to o}(\Sigma)$, and $\boldsymbol{C} \in \mathrm{cwff}_\alpha(\Sigma)$ (for any type α):*

$\overline{\nabla}_\neg$: *$\neg\boldsymbol{A} \in \mathcal{H}$ iff $\boldsymbol{A} \notin \mathcal{H}$.*
$\overline{\nabla}_\vee$: *$(\boldsymbol{A} \vee \boldsymbol{B}) \in \mathcal{H}$ iff $\boldsymbol{A} \in \mathcal{H}$ or $\boldsymbol{B} \in \mathcal{H}$.*
$\overline{\nabla}_\forall$: *$(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$ if and only if $\boldsymbol{FD} \in \mathcal{H}$ for every $\boldsymbol{D} \in \mathrm{cwff}_\alpha(\Sigma)$.*
$\overline{\nabla}_\forall^\beta$: *$(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$ iff $(\boldsymbol{FD}){\downarrow_\beta} \in \mathcal{H}$ for every $\boldsymbol{D} \in \mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta}$.*
$\overline{\nabla}_r$: *$(\boldsymbol{C} \doteq^\alpha \boldsymbol{C}) \in \mathcal{H}$.*

Proof. If $\neg\boldsymbol{A} \in \mathcal{H}$, then $\boldsymbol{A} \notin \mathcal{H}$ by $\vec{\nabla}_c$. If $\boldsymbol{A} \notin \mathcal{H}$, then $\neg\boldsymbol{A} \in \mathcal{H}$ since $\mathcal{H}$ is saturated. So, $\overline{\nabla}_\neg$ holds.

If $(\boldsymbol{A} \vee \boldsymbol{B}) \in \mathcal{H}$, then $\boldsymbol{A} \in \mathcal{H}$ or $\boldsymbol{B} \in \mathcal{H}$ by $\vec{\nabla}_\vee$. We prove the converse by contraposition. Suppose $(\boldsymbol{A} \vee \boldsymbol{B}) \notin \mathcal{H}$. By saturation we have $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \mathcal{H}$, and by $\vec{\nabla}_\wedge$ we get $\neg\boldsymbol{A} \in \mathcal{H}$ and $\neg\boldsymbol{B} \in \mathcal{H}$. So, by $\vec{\nabla}_c$, $\boldsymbol{A} \notin \mathcal{H}$ and $\boldsymbol{B} \notin \mathcal{H}$. Thus, $\overline{\nabla}_\vee$ holds.

One direction of $\overline{\nabla}_\forall$ is $\vec{\nabla}_\forall$. For one direction of $\overline{\nabla}_\forall^\beta$, note that if $(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$, then for any $\boldsymbol{D} \in \mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta}$ we have $(\boldsymbol{FD}){\downarrow_\beta} \in \mathcal{H}$ by $\vec{\nabla}_\forall$ and $\vec{\nabla}_\beta$.

Suppose $(\Pi^\alpha \boldsymbol{F}) \notin \mathcal{H}$. By saturation, $\neg(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$. By $\vec{\nabla}_\exists$, there is a parameter $w_\alpha \in \Sigma_\alpha$ such that $\neg(\boldsymbol{F}w) \in \mathcal{H}$. By $\vec{\nabla}_c$, we know $(\boldsymbol{F}w) \notin \mathcal{H}$. This shows the other direction of $\overline{\nabla}_\forall$. Furthermore, by $\vec{\nabla}_\beta$ we know $\neg(\boldsymbol{F}w){\downarrow_\beta} \in \mathcal{H}$ and so $(\boldsymbol{F}w){\downarrow_\beta} \notin \mathcal{H}$. Since w is β-normal, we also have the other direction of $\overline{\nabla}_\forall^\beta$.

Finally, $\overline{\nabla}_r$ follows directly from saturation and $\vec{\nabla}^r_{\dot{=}}$. ⊣

LEMMA 6.26 (Saturated sets lemma for $\mathfrak{b}$). *Suppose $\mathcal{H} \in \mathfrak{Hint}_*$ where $* \in \{\beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. If $\mathcal{H}$ is saturated, then the following property holds for all $\boldsymbol{A}, \boldsymbol{B} \in$* $\mathrm{cwff}_o(\Sigma)$.

$\overline{\nabla}_{\mathfrak{b}}$: *$\boldsymbol{A} \doteq^o \boldsymbol{B} \in \mathcal{H}$ or $\boldsymbol{A} \doteq^o \neg\boldsymbol{B} \in \mathcal{H}$.*

PROOF. Suppose $(\boldsymbol{A} \doteq^o \boldsymbol{B}) \notin \mathcal{H}$ and $(\boldsymbol{A} \doteq^o \neg\boldsymbol{B}) \notin \mathcal{H}$. By saturation, $\neg(\boldsymbol{A} \doteq^o \boldsymbol{B}) \in \mathcal{H}$ and $\neg(\boldsymbol{A} \doteq^o \neg\boldsymbol{B}) \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{b}}$, we must have $\{\boldsymbol{A}, \neg\boldsymbol{B}\} \subseteq \mathcal{H}$ or $\{\neg\boldsymbol{A}, \boldsymbol{B}\} \subseteq \mathcal{H}$. We must also have $\{\boldsymbol{A}, \neg\neg\boldsymbol{B}\} \subseteq \mathcal{H}$ or $\{\neg\boldsymbol{A}, \neg\boldsymbol{B}\} \subseteq \mathcal{H}$. Each of the four cases leads to an immediate contradiction to $\vec{\nabla}_c$. ⊣

LEMMA 6.27 (Saturated sets lemma for η). *Suppose $\mathcal{H} \in \mathfrak{Hint}_*$ where $* \in \{\beta\eta, \beta\eta\mathfrak{b}\}$. If $\mathcal{H}$ is saturated, then the following property holds for every type α and $\boldsymbol{A} \in$* $\mathrm{cwff}_\alpha(\Sigma)$:

$\overline{\nabla}_\eta$: $(\boldsymbol{A} \doteq^\alpha \boldsymbol{A}{\downarrow_{\beta\eta}}) \in \mathcal{H}$.

PROOF. If $(\boldsymbol{A} \doteq \boldsymbol{A}{\downarrow_{\beta\eta}}) \notin \mathcal{H}$, then by saturation $\neg(\boldsymbol{A} \doteq \boldsymbol{A}{\downarrow_{\beta\eta}}) \in \mathcal{H}$. So, by $\vec{\nabla}_\eta$ we have $\neg(\boldsymbol{A}{\downarrow_{\beta\eta}} \doteq^\alpha \boldsymbol{A}{\downarrow_{\beta\eta}}) \in \mathcal{H}$. But this contradicts $\vec{\nabla}^r_{\dot{=}}$. ⊣

LEMMA 6.28 (Saturated sets lemma for ξ). *Suppose $\mathcal{H} \in \mathfrak{Hint}_*$ where $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$. If $\mathcal{H}$ is saturated, then the following properties hold for all $\alpha, \beta \in \mathcal{T}$ and* $(\lambda X_\alpha.\boldsymbol{M}), (\lambda X.\boldsymbol{N}) \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma)$:

$\overline{\nabla}_\xi$: $(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \mathcal{H}$ *iff* $([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N}) \in \mathcal{H}$ *for every* $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)$.

$\overline{\nabla}^\beta_\xi$: $(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \mathcal{H}$ *iff* $([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N}){\downarrow_\beta} \in \mathcal{H}$ *for every* $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta}$.

PROOF. Suppose $(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \mathcal{H}$ and $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)$. We can apply $\vec{\nabla}_\forall$ and $\vec{\nabla}_\beta$ using the closed formula $(\lambda K_{\alpha\to\beta}.[\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta K\boldsymbol{A})$ to obtain

$$(\neg([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{M}) \vee [\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N}) \in \mathcal{H}.$$

Since $\neg([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{M}) \notin \mathcal{H}$ (by $\vec{\nabla}^r_{\dot{=}}$), we know $([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N}) \in \mathcal{H}$. This shows one direction of $\overline{\nabla}_\xi$. By $\vec{\nabla}_\beta$ we have $([\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N}){\downarrow_\beta} \in \mathcal{H}$. Since this holds in particular for any $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma){\downarrow_\beta}$, this shows one direction of $\overline{\nabla}^\beta_\xi$.

Suppose $(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \notin \mathcal{H}$. We show that there is a (β-normal) $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)$ with $[\boldsymbol{A}/X]\boldsymbol{M} \doteq^\beta [\boldsymbol{A}/X]\boldsymbol{N} \notin \mathcal{H}$. By saturation, $\neg(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \mathcal{H}$. By $\vec{\nabla}_\xi$, there is a parameter $w_\alpha \in \Sigma_\alpha$ such that $\neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N}) \in \mathcal{H}$. By $\vec{\nabla}_c$, $[w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N} \notin \mathcal{H}$. Choosing $\boldsymbol{A} := w$ we have the other direction of $\overline{\nabla}_\xi$. Since w is β-normal and $([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N}){\downarrow_\beta} \notin \mathcal{H}$ (using $\vec{\nabla}_\beta$), we have the other direction of $\overline{\nabla}^\beta_\xi$. ⊣

LEMMA 6.29 (Saturated sets lemma for $\mathfrak{f}$). *Suppose $\mathcal{H} \in \mathfrak{Hint}_*$ where $* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$. If $\mathcal{H}$ is saturated, then the following property holds for any types α and β and* $\boldsymbol{G}, \boldsymbol{H} \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma)$.

$\overline{\nabla}_{\mathfrak{f}}$: $\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H} \in \mathcal{H}$ *iff* $\boldsymbol{G}\boldsymbol{A} \doteq^\beta \boldsymbol{H}\boldsymbol{A} \in \mathcal{H}$ *for every* $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)$.

$\overline{\nabla}_{\mathfrak{f}}^{\beta}$: *$\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H} \in \mathscr{H}$ iff $(\boldsymbol{GA} \doteq^{\beta} \boldsymbol{HA}){\downarrow}_{\beta} \in \mathscr{H}$ for every $\boldsymbol{A} \in \mathrm{cwff}_{\alpha}(\Sigma){\downarrow}_{\beta}$.*

PROOF. Suppose $(\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H}) \in \mathscr{H}$ and $\boldsymbol{A} \in \mathrm{cwff}_{\alpha}(\Sigma)$. Since $(\boldsymbol{A} \doteq^{\alpha} \boldsymbol{A}) \in \mathscr{H}$ by $\overline{\nabla}_r$ we have $(\boldsymbol{GA} \doteq^{\beta} \boldsymbol{HA}) \in \mathscr{H}$ by $\vec{\nabla}_{\doteq}^{\to}$ (cf. Lemma 6.23). This shows one direction of $\overline{\nabla}_{\mathfrak{f}}$. By $\vec{\nabla}_{\beta}$ we have $(\boldsymbol{GA} \doteq^{\beta} \boldsymbol{HA}){\downarrow}_{\beta} \in \mathscr{H}$. Since this holds in particular for any $\boldsymbol{A} \in \mathrm{cwff}_{\alpha}(\Sigma){\downarrow}_{\beta}$, this shows one direction of $\overline{\nabla}_{\mathfrak{f}}^{\beta}$.

Suppose $(\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H}) \notin \mathscr{H}$. By saturation, $\neg(\boldsymbol{G} \doteq^{\alpha\to\beta} \boldsymbol{H}) \in \mathscr{H}$. By $\vec{\nabla}_{\mathfrak{f}}$, there is a parameter $w_{\alpha} \in \Sigma_{\alpha}$ such that $\neg(\boldsymbol{G}w \doteq^{\beta} \boldsymbol{H}w) \in \mathscr{H}$. By $\vec{\nabla}_c$, $(\boldsymbol{G}w \doteq^{\beta} \boldsymbol{H}w) \notin \mathscr{H}$. Choosing $\boldsymbol{A} := w$ we have the other direction of $\overline{\nabla}_{\mathfrak{f}}$. Since w is β-normal and $(\boldsymbol{G}w \doteq^{\beta} \boldsymbol{H}w){\downarrow}_{\beta} \notin \mathscr{H}$ (using $\vec{\nabla}_{\beta}$), we have the other direction of $\overline{\nabla}_{\mathfrak{f}}^{\beta}$. ⊣

In Lemma 3.24, we compared properties η, ξ and $\mathfrak{f}$ of models by showing $\mathfrak{f}$ is equivalent to η plus ξ. Similarly, Theorem 6.31 compares $\vec{\nabla}_{\eta}$, $\vec{\nabla}_{\xi}$, and $\vec{\nabla}_{\mathfrak{f}}$ as properties of Hintikka sets. Showing $\vec{\nabla}_{\mathfrak{f}}$ implies $\vec{\nabla}_{\eta}$ requires saturation and must be shown in several steps reflected by Lemma 6.30.

LEMMA 6.30. *Let $\mathscr{H}$ be a saturated Hintikka set satisfying $\vec{\nabla}_{\mathfrak{f}}$.*

(1) *For all $\boldsymbol{F} \in \mathrm{cwff}_{\alpha\to\beta}$ we have $(\lambda X_{\alpha}.\boldsymbol{F}X) \doteq^{\alpha\to\beta} \boldsymbol{F} \in \mathscr{H}$.*
(2) *For all $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_{\alpha}(\Sigma)$, if $\boldsymbol{A}$ η-reduces to $\boldsymbol{B}$ in one step, then $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{B} \in \mathscr{H}$.*
(3) *For all $\boldsymbol{A} \in \mathrm{cwff}_{\alpha}(\Sigma)$, $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$.*
(4) *For all $\boldsymbol{A} \in \mathrm{cwff}_{o}(\Sigma)$, if $\boldsymbol{A} \in \mathscr{H}$, then $\boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$.*

PROOF. To show part (1), suppose $(\lambda X_{\alpha}.\boldsymbol{F}X) \doteq^{\alpha\to\beta} \boldsymbol{F} \notin \mathscr{H}$. By saturation, $\neg((\lambda X_{\alpha}.\boldsymbol{F}X) \doteq^{\alpha\to\beta} \boldsymbol{F}) \in \mathscr{H}$. By $\vec{\nabla}_{\mathfrak{f}}$, there is a parameter w_{α} such that

$$\neg(((\lambda X_{\alpha}.\boldsymbol{F}X)w) \doteq^{\beta} (\boldsymbol{F}w)) \in \mathscr{H}.$$

By $\vec{\nabla}_{\beta}$, $\neg((\boldsymbol{F}w) \doteq^{\beta} (\boldsymbol{F}w)) \in \mathscr{H}$, which contradicts $\vec{\nabla}_{\doteq}^{r}$ (cf. Lemma 6.23).

We prove part (2) by induction on the position of the η-redex in $\boldsymbol{A}$. If $\boldsymbol{A}$ is the η-redex reduced to obtain $\boldsymbol{B}$, then this follows from part (1). Suppose $\boldsymbol{A} \equiv (\boldsymbol{F}_{\gamma\to\alpha}\boldsymbol{C}_{\gamma})$ and $\boldsymbol{B} \equiv (\boldsymbol{G}_{\gamma\to\alpha}\boldsymbol{C})$ where $\boldsymbol{F}$ η-reduces to $\boldsymbol{G}$ in one step. By induction, we know $\boldsymbol{F} \doteq^{\gamma\to\alpha} \boldsymbol{G} \in \mathscr{H}$. By $\overline{\nabla}_r$, $\boldsymbol{C} \doteq^{\gamma} \boldsymbol{C} \in \mathscr{H}$. By $\vec{\nabla}_{\doteq}^{\to}$, we have $(\boldsymbol{FC}) \doteq^{\alpha} (\boldsymbol{GC}) \in \mathscr{H}$ as desired. The case in which $\boldsymbol{A} \equiv (\boldsymbol{F}_{\gamma\to\alpha}\boldsymbol{C}_{\gamma})$ and $\boldsymbol{B} \equiv (\boldsymbol{F}\boldsymbol{D}_{\gamma})$ where $\boldsymbol{C}$ η-reduces to $\boldsymbol{D}$ in one step is analogous.

Suppose $\boldsymbol{A} \equiv (\lambda Y_{\beta}.\boldsymbol{C}_{\gamma})$ and $\boldsymbol{B} \equiv (\lambda Y_{\beta}.\boldsymbol{D}_{\gamma})$ where $\boldsymbol{C}$ η-reduces to $\boldsymbol{D}$ in one step. Let p be the position of the redex in $\boldsymbol{C}$. Assume $\boldsymbol{A} \doteq^{\beta\to\gamma} \boldsymbol{B} \notin \mathscr{H}$. By saturation, $\neg(\boldsymbol{A} \doteq^{\beta\to\gamma} \boldsymbol{B}) \in \mathscr{H}$. By $\vec{\nabla}_{\mathfrak{f}}$, there is some parameter w_{β} such that $\neg(\boldsymbol{A}w \doteq^{\gamma} \boldsymbol{B}w) \in \mathscr{H}$. By $\vec{\nabla}_{\beta}$, we know $\neg([w/Y]\boldsymbol{C} \doteq^{\gamma} [w/Y]\boldsymbol{D}) \in \mathscr{H}$. Note that $[w/Y]\boldsymbol{C}$ η-reduces to $[w/Y]\boldsymbol{D}$ in one step by reducing the redex at position p in $[w/Y]\boldsymbol{C}$. So, by the induction hypothesis, $[w/Y]\boldsymbol{C} \doteq^{\gamma} [w/Y]\boldsymbol{D} \in \mathscr{H}$, contradicting $\vec{\nabla}_c$.

Part (3) follows by induction on the number of $\beta\eta$-reductions from $\boldsymbol{A}$ to $\boldsymbol{A}{\downarrow}_{\beta\eta}$. If $\boldsymbol{A}$ is $\beta\eta$-normal, we have $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{A} \in \mathscr{H}$ by $\overline{\nabla}_r$. If $\boldsymbol{A}$ reduces to $\boldsymbol{A}{\downarrow}_{\beta\eta}$ in $n+1$ steps, then there is some $\boldsymbol{B}_{\alpha}$ such that $\boldsymbol{A}$ reduces to $\boldsymbol{B}$ in one step and $\boldsymbol{B}$ reduces to $\boldsymbol{A}{\downarrow}_{\beta\eta}$ in n steps. By induction, we have $\boldsymbol{B} \doteq^{\alpha} \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$. If $\boldsymbol{A}$ β-reduces to $\boldsymbol{B}$ in one step, then $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{B} \in \mathscr{H}$ by $\overline{\nabla}_r$ and $\vec{\nabla}_{\beta}$. If $\boldsymbol{A}$ η-reduces to $\boldsymbol{B}$ in one step, then $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{B} \in \mathscr{H}$

by part (2). Using $\vec{\nabla}^{tr}_{\doteq}$, $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{B} \in \mathscr{H}$ and $\boldsymbol{B} \doteq^{\alpha} \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$ imply $\boldsymbol{A} \doteq^{\alpha} \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$ as desired.

Finally, to show part (4), suppose $\boldsymbol{A} \in \mathscr{H}$. By part (3), $\boldsymbol{A} \doteq^{o} \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$. By $\vec{\nabla}_{\forall}$, $\neg(\lambda X_o\boldsymbol{.}X)\boldsymbol{A} \vee (\lambda X_o\boldsymbol{.}X)\ \boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$. By $\vec{\nabla}_{\beta}$ and $\vec{\nabla}_{\vee}$, we have $\neg\boldsymbol{A} \in \mathscr{H}$ (contradicting $\vec{\nabla}_{c}$) or $\boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$. Hence, $\boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$. ⊣

THEOREM 6.31. *Let $\mathscr{H}$ be a Hintikka set.*

(1) *If $\mathscr{H}$ satisfies $\vec{\nabla}_{\eta}$ and $\vec{\nabla}_{\xi}$, then $\mathscr{H}$ satisfies $\vec{\nabla}_{\mathfrak{f}}$.*
(2) *If $\mathscr{H}$ satisfies $\vec{\nabla}_{\mathfrak{f}}$, then $\mathscr{H}$ satisfies $\vec{\nabla}_{\xi}$.*
(3) *If $\mathscr{H}$ is saturated and satisfies $\vec{\nabla}_{\mathfrak{f}}$, then $\mathscr{H}$ satisfies $\vec{\nabla}_{\eta}$.*

PROOF. Suppose $\mathscr{H}$ satisfies $\vec{\nabla}_{\eta}$ and $\vec{\nabla}_{\xi}$. Assume $\neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G}) \in \mathscr{H}$. By $\vec{\nabla}_{\eta}$, $\neg((\lambda X_{\alpha}\boldsymbol{.F}X) \doteq^{\alpha\to\beta} (\lambda X\boldsymbol{.G}X)) \in \mathscr{H}$. By $\vec{\nabla}_{\xi}$, there is a parameter w_{α} such that $\neg((\boldsymbol{F}w) \doteq^{\beta} (\boldsymbol{G}w)) \in \mathscr{H}$. Thus, $\vec{\nabla}_{\mathfrak{f}}$ holds.

Suppose $\mathscr{H}$ satisfies $\vec{\nabla}_{\mathfrak{f}}$ and $\neg(\lambda X_{\alpha}\boldsymbol{.M} \doteq^{\alpha\to\beta} \lambda X\boldsymbol{.N}) \in \mathscr{H}$. By $\vec{\nabla}_{\mathfrak{f}}$, there is a parameter w_{α} such that $\neg((\lambda X_{\alpha}\boldsymbol{.M})w \doteq^{\beta} (\lambda X\boldsymbol{.N})w) \in \mathscr{H}$. By $\vec{\nabla}_{\beta}$, we have $\neg([w/X]\boldsymbol{M} \doteq^{\beta} [w/X]\boldsymbol{N}) \in \mathscr{H}$. Thus, $\vec{\nabla}_{\xi}$ holds.

Suppose $\mathscr{H}$ is saturated and satisfies $\vec{\nabla}_{\mathfrak{f}}$. Assume $\boldsymbol{A} \in \mathscr{H}$, $\boldsymbol{B} \in \mathrm{cwff}_o(\Sigma)$, $\boldsymbol{A} \equiv_{\beta\eta} \boldsymbol{B}$ and $\boldsymbol{B} \notin \mathscr{H}$. By saturation, we know $\neg\boldsymbol{B} \in \mathscr{H}$. By Lemma 6.30(4), we know $\boldsymbol{A}{\downarrow}_{\beta\eta} \in \mathscr{H}$ and $\neg\ \boldsymbol{B}{\downarrow}_{\beta\eta} \in \mathscr{H}$. Since $\boldsymbol{A}{\downarrow}_{\beta\eta} \equiv \boldsymbol{B}{\downarrow}_{\beta\eta}$, this contradicts $\vec{\nabla}_{c}$. ⊣

6.3. Model existence theorems. We shall now present the proof of the abstract extension lemma, which will nearly immediately yield the model existence theorems. For the proof we adapt the construction of Henkin's completeness proof from [26, 27].

LEMMA 6.32 (Abstract extension lemma). *Let Σ be a signature, Γ_{Σ} be a compact abstract consistency class in $\mathfrak{Acc}_*$, where $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$, and let $\Phi \in \Gamma_{\Sigma}$ be sufficiently Σ-pure. Then there exists a Σ-Hintikka set $\mathscr{H} \in \mathfrak{Hint}_*$, such that $\Phi \subseteq \mathscr{H}$. Furthermore, if Γ_{Σ} is saturated, then $\mathscr{H}$ is saturated.*

PROOF. In the following argument, note that α, β, and γ are types as usual, while δ, ε, σ and τ are ordinals.

By Remark 3.16, there is an infinite cardinal $\aleph_s$ which is the cardinality of Σ_{α} for each type α. This easily implies $\mathrm{cwff}_{\alpha}(\Sigma)$ is of cardinality $\aleph_s$ for each type α. Let ε be the first ordinal of this cardinality. (In the countable case, ε is ω.) Since the cardinality of $\mathrm{cwff}_o(\Sigma)$ is $\aleph_s$, we can use the well-ordering principle to enumerate $\mathrm{cwff}_o(\Sigma)$ as $(\boldsymbol{A}^{\delta})_{\delta<\varepsilon}$.

Let α be a type. For each $\delta < \varepsilon$, let U^{δ}_{α} be the set of constants of type α which occur in a sentence in the set $\{\,\boldsymbol{A}^{\sigma} \mid \sigma \leq \delta\,\}$. Since $\delta < \varepsilon$, the set $\{\,\boldsymbol{A}^{\sigma} \mid \sigma \leq \delta\,\}$ has cardinality less than $\aleph_s$. Hence, U^{δ}_{α} has cardinality less than $\aleph_s$. By sufficient purity, we know there is a set of parameters $P_{\alpha} \subseteq \Sigma_{\alpha}$ of cardinality $\aleph_s$ such that the parameters in P_{α} do not occur in the sentences of Φ. So, $P_{\alpha} \setminus U^{\delta}_{\alpha}$ must have cardinality $\aleph_s$ for any $\delta < \varepsilon$. Using the axiom of choice, we can find a sequence $(w^{\delta}_{\alpha})_{\delta<\varepsilon}$ where for each $\delta < \varepsilon$, $w^{\delta}_{\alpha} \in P_{\alpha} \setminus (U^{\delta}_{\alpha} \cup \{\,w^{\sigma}_{\alpha} \mid \sigma < \delta\,\})$. That is, for each type α, we know w^{δ}_{α} is a parameter of type α which does not occur in any sentence in $\Phi \cup \{\,\boldsymbol{A}^{\sigma} \mid \sigma \leq \delta\,\}$. As a consequence, if w^{δ}_{α} occurs in $\boldsymbol{A}^{\sigma}$, then $\delta < \sigma$. Also, we have ensured that if $w^{\delta}_{\alpha} \equiv w^{\sigma}_{\alpha}$, then $\delta \equiv \sigma$ for any $\delta, \sigma < \varepsilon$.

The parameters w_α^δ are intended to serve as witnesses. To ease the argument, we define two sequences of witnessing sentences related to the sequence $(\boldsymbol{A}^\delta)_{\delta<\varepsilon}$. For each $\delta < \varepsilon$, let $\boldsymbol{E}^\delta := \neg(\boldsymbol{B} w_\alpha^\delta)$ if $\boldsymbol{A}^\delta$ is of the form $\neg(\Pi^\alpha \boldsymbol{B})$, and let $\boldsymbol{E}^\delta := \boldsymbol{A}^\delta$ otherwise. If $* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$ and $\boldsymbol{A}^\delta$ is of the form $\neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G})$, let $\boldsymbol{X}^\delta := \neg(\boldsymbol{F} w_\alpha^\delta \doteq^\beta \boldsymbol{G} w_\alpha^\delta)$. If $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$ and $\boldsymbol{A}^\delta$ is of the form $\neg((\lambda X_\alpha . \boldsymbol{M}) \doteq^{\alpha\to\beta} (\lambda X . \boldsymbol{N}))$, let $\boldsymbol{X}^\delta := \neg([w_\alpha^\delta/X]\boldsymbol{M} \doteq^\beta [w_\alpha^\delta/X]\boldsymbol{N})$. Otherwise, let $\boldsymbol{X}^\delta := \boldsymbol{A}^\delta$. (Notice that any sentence $\neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G})$ is also of the form $\neg(\Pi^\gamma \boldsymbol{B})$, where γ is $(\alpha \to \beta) \to o$. So, whenever $\boldsymbol{X}^\delta \not\equiv \boldsymbol{A}^\delta$, we must also have $\boldsymbol{E}^\delta \not\equiv \boldsymbol{A}^\delta$.)

We construct $\mathscr{H}$ by inductively constructing a transfinite sequence $(\mathscr{H}^\delta)_{\delta<\varepsilon}$ such that $\mathscr{H}^\delta \in \Gamma_\Sigma$ for each $\delta < \varepsilon$. Then the Σ-Hintikka set is $\mathscr{H} := \bigcup_{\delta<\varepsilon} \mathscr{H}^\delta$. We define $\mathscr{H}^0 := \Phi$. For limit ordinals δ, we define $\mathscr{H}^\delta := \bigcup_{\sigma<\delta} \mathscr{H}^\sigma$.

In the successor case, if $\mathscr{H}^\delta * \boldsymbol{A}^\delta \in \Gamma_\Sigma$, then we let $\mathscr{H}^{\delta+1} := \mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta * \boldsymbol{X}^\delta$. If $\mathscr{H}^\delta * \boldsymbol{A}^\delta \notin \Gamma_\Sigma$, we let $\mathscr{H}^{\delta+1} := \mathscr{H}^\delta$.

We show by induction that for every $\delta < \varepsilon$, type α and parameter w_α^τ which occurs in some sentence in $\mathscr{H}^\delta$, we have $\tau < \delta$. The base case holds since no w_α^τ occurs in any sentence in $\mathscr{H}^0 \equiv \Phi$. For any limit ordinal δ, if w_α^τ occurs in some sentence in $\mathscr{H}^\delta$, then by definition of $\mathscr{H}^\delta$, w_α^τ already occurs in some sentence in $\mathscr{H}^\sigma$ for some $\sigma < \delta$. So, $\tau < \sigma < \delta$.

For any successor ordinal $\delta + 1$, suppose w_α^τ occurs in some sentence in $\mathscr{H}^{\delta+1}$. If it already occurred in a sentence in $\mathscr{H}^\delta$, then we have $\tau < \delta < \delta + 1$ by the inductive assumption. So, we need only consider the case where w_α^τ occurs in a sentence in $\mathscr{H}^{\delta+1} \setminus \mathscr{H}^\delta$. Note that $(\mathscr{H}^{\delta+1} \setminus \mathscr{H}^\delta) \subseteq \{\boldsymbol{A}^\delta, \boldsymbol{E}^\delta, \boldsymbol{X}^\delta\}$. In any case, note that if τ is δ, then we are done, since $\delta < \delta + 1$. If w_α^τ is any parameter with $\tau \not\equiv \delta$ and occurs in $\boldsymbol{E}^\delta$ or $\boldsymbol{X}^\delta$, then it must also occur in $\boldsymbol{A}^\delta$ (by noting that $w_\alpha^\tau \not\equiv w_\alpha^\delta$ and inspecting the possible definitions of $\boldsymbol{E}^\delta$ and $\boldsymbol{X}^\delta$), in which case $\tau < \delta < \delta + 1$.

In particular, we now know w_α^δ does not occur in any sentence of $\mathscr{H}^\delta$ for any $\delta < \varepsilon$ and type α.

Next we show by induction that $\mathscr{H}^\delta \in \Gamma_\Sigma$ for all $\delta < \varepsilon$. The base case holds by the assumption that $\mathscr{H}^0 \equiv \Phi \in \Gamma_\Sigma$. For any limit ordinal δ, assume $\mathscr{H}^\sigma \in \Gamma_\Sigma$ for every $\sigma < \delta$. We have $\mathscr{H}^\delta \equiv \bigcup_{\sigma<\delta} \mathscr{H}^\sigma \in \Gamma_\Sigma$ by compactness, since any finite subset of $\mathscr{H}^\delta$ is a subset of $\mathscr{H}^\sigma$ for some $\sigma < \delta$.

For any successor ordinal $\delta + 1$, we assume $\mathscr{H}^\delta \in \Gamma_\Sigma$. We have to show that $\mathscr{H}^{\delta+1} \in \Gamma_\Sigma$. This is trivial in case $\mathscr{H}^\delta * \boldsymbol{A}^\delta \notin \Gamma_\Sigma$ (for all abstract consistency classes) since $\mathscr{H}^{\delta+1} \equiv \mathscr{H}^\delta$. Suppose $\mathscr{H}^\delta * \boldsymbol{A}^\delta \in \Gamma_\Sigma$. We consider three sub-cases:

(i) If $\boldsymbol{E}^\delta \equiv \boldsymbol{A}^\delta$ and $\boldsymbol{X}^\delta \equiv \boldsymbol{A}^\delta$, then $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta * \boldsymbol{X}^\delta \in \Gamma_\Sigma$ since $\mathscr{H}^\delta * \boldsymbol{A}^\delta \in \Gamma_\Sigma$.

(ii) If $\boldsymbol{E}^\delta \not\equiv \boldsymbol{A}^\delta$ and $\boldsymbol{X}^\delta \equiv \boldsymbol{A}^\delta$, then $\boldsymbol{A}^\delta$ is of the form $\neg\Pi^\alpha \boldsymbol{B}$ and $\boldsymbol{E}^\delta \equiv \neg\boldsymbol{B} w_\alpha^\delta$. We conclude that $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta \in \Gamma_\Sigma$ by $\nabla_\exists$ since w_α^δ does not occur in $\boldsymbol{A}^\delta$ or any sentence of $\mathscr{H}^\delta$. Since $\boldsymbol{X}^\delta \equiv \boldsymbol{A}^\delta$, this is the same as concluding $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta * \boldsymbol{X}^\delta \in \Gamma_\Sigma$.

(iii) If $\boldsymbol{X}^\delta \not\equiv \boldsymbol{A}^\delta$, then $* \in \{\beta\xi, \beta\mathfrak{f}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$ (by the definition of $\boldsymbol{X}^\delta$). $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta \in \Gamma_\Sigma$ by $\nabla_\exists$ since $w_{(\alpha\to\beta)\to o}^\delta$ does not occur in $\boldsymbol{A}^\delta$ or any sentence in $\mathscr{H}^\delta$. Now, w_α^δ (which is different from $w_{(\alpha\to\beta)\to o}^\delta$ since it has a different type) does not occur in any sentence in $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta$. We have $\mathscr{H}^\delta * \boldsymbol{A}^\delta * \boldsymbol{E}^\delta * \boldsymbol{X}^\delta \in \mathscr{H}$ by ∇_ξ (if $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$) or by $\nabla_\mathfrak{f}$ (if $* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$).

Since Γ_Σ is compact, we also have $\mathscr{H} \in \Gamma_\Sigma$.

Now we know that our inductively defined set $\mathcal{H}$ is indeed in Γ_Σ and that $\Phi \subseteq \mathcal{H}$. In order to apply Lemma 6.21, we must check $\mathcal{H}$ is maximal, satisfies $\vec{\nabla}_\exists$, $\vec{\nabla}_\xi$ (if $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$), and $\vec{\nabla}_\mathfrak{f}$ (if $* \in \{\beta\mathfrak{f}, \beta\mathfrak{fb}\}$). It is immediate from the construction that $\vec{\nabla}_\exists$ holds since if $\neg(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$, then $\neg(\boldsymbol{F} w_\alpha^\delta) \in \mathcal{H}$ where δ is the ordinal such that $\boldsymbol{A}^\delta \equiv \neg(\Pi^\alpha \boldsymbol{F})$. If $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$, then we have ensured $\vec{\nabla}_\xi$ holds since $\neg([w_\alpha^\delta/X]\boldsymbol{M} \doteq^\beta [w_\alpha^\delta/X]\boldsymbol{N}) \in \mathcal{H}$ whenever $\neg((\lambda X_\alpha . \boldsymbol{M}) \doteq^{\alpha\to\beta} (\lambda X . \boldsymbol{N})) \in \mathcal{H}$ where δ is the ordinal such that $\boldsymbol{A}^\delta \equiv \neg((\lambda X_\alpha . \boldsymbol{M}) \doteq^{\alpha\to\beta} (\lambda X . \boldsymbol{N}))$. Similarly, we have ensured $\vec{\nabla}_\mathfrak{f}$ holds when $* \in \{\beta\mathfrak{f}, \beta\mathfrak{fb}\}$ since $\neg(\boldsymbol{F} w_\alpha^\delta \doteq^\beta \boldsymbol{G} w_\alpha^\delta) \in \mathcal{H}$ whenever $\neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G}) \in \mathcal{H}$ where δ is the ordinal such that $\boldsymbol{A}^\delta \equiv \neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G})$.

It only remains to show that $\mathcal{H}$ is maximal in Γ_Σ. So, let $\boldsymbol{A} \in \mathrm{cwff}_o$ and $\mathcal{H} * \boldsymbol{A} \in \Gamma_\Sigma$ be given. Note that $\boldsymbol{A} \equiv \boldsymbol{A}^\delta$ for some $\delta < \varepsilon$. Since $\mathcal{H}$ is closed under subsets we know that $\mathcal{H}^\delta * \boldsymbol{A}^\delta \in \Gamma_\Sigma$. By definition of $\mathcal{H}^{\delta+1}$ we conclude that $\boldsymbol{A}^\delta \in \mathcal{H}^{\delta+1}$ and hence $\boldsymbol{A} \in \mathcal{H}$.

So, Lemma 6.21 implies $\mathcal{H} \in \mathfrak{Hint}_*$ and $\mathcal{H}$ is saturated if Γ_Σ is saturated. ⊣

We now use the Σ-Hintikka sets, guaranteed by Lemma 6.32, to construct a Σ-valuation for the Σ-term evaluation that turns it into a model.

THEOREM 6.33 (Model existence theorem for saturated sets). *For all* $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{fb}\}$ *we have*: *If* $\mathcal{H}$ *is a saturated Hintikka set in* $\mathfrak{Hint}_*$ (*cf. Definition* 6.20), *then there exists a model* $\mathcal{M} \in \mathfrak{M}_*$ (*cf. Definition* 3.49) *that satisfies* $\mathcal{H}$. *Furthermore, each domain* $\mathcal{D}_\alpha$ *of* $\mathcal{M}$ *has cardinality at most* $\aleph_s$.

PROOF. We start with the construction of a Σ-model $\mathcal{M}_1^{\mathcal{H}}$ for $\mathcal{H}$ based on the term evaluation $\mathcal{TE}(\Sigma)^\beta$. This model may not be in the model class $\mathfrak{M}_*$ as it may not satisfy property $\mathfrak{q}$. However, we will be able to use Theorem 3.62 to obtain a model of $\mathcal{H}$ which is.

Note that since $\mathcal{H}$ is saturated, by Lemma 6.25, $\mathcal{H}$ satisfies $\overline{\nabla}_\neg$, $\overline{\nabla}_\vee$, and $\overline{\nabla}_\forall^\beta$.

The domain of type α of the evaluation $\mathcal{TE}(\Sigma)^\beta$ (cf. Definition 3.35 and Lemma 3.36) is $\mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$, which has cardinality $\aleph_s$. To construct $\mathcal{M}_1^{\mathcal{H}}$, we simply need to give a valuation function for this evaluation. This valuation function should be a function $\upsilon\colon \mathrm{cwff}_o(\Sigma)\big\downarrow_\beta \longrightarrow \{\mathtt{T}, \mathtt{F}\}$. We define

$$\upsilon(\boldsymbol{A}) := \begin{cases} \mathtt{T} & \text{if } \boldsymbol{A} \in \mathcal{H}, \\ \mathtt{F} & \text{if } \boldsymbol{A} \notin \mathcal{H}. \end{cases}$$

To show υ is a valuation, we must check the logical constants are interpreted appropriately. For each $\boldsymbol{A} \in \mathrm{cwff}_o(\Sigma)\big\downarrow_\beta$, we have $\upsilon(\neg\boldsymbol{A}) \equiv \mathtt{T}$ iff $\upsilon(\boldsymbol{A}) \equiv \mathtt{F}$ since $\neg\boldsymbol{A} \in \mathcal{H}$ iff $\boldsymbol{A} \notin \mathcal{H}$ by $\overline{\nabla}_\neg$. For each $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_o(\Sigma)\big\downarrow_\beta$, we have $\upsilon(\boldsymbol{A} \vee \boldsymbol{B}) \equiv \mathtt{T}$ iff $\upsilon(\boldsymbol{A}) \equiv \mathtt{T}$ or $\upsilon(\boldsymbol{B}) \equiv \mathtt{T}$, since $(\boldsymbol{A} \vee \boldsymbol{B}) \in \mathcal{H}$ iff $\boldsymbol{A} \in \mathcal{H}$ or $\boldsymbol{B} \in \mathcal{H}$ by $\overline{\nabla}_\vee$. Finally, for each type α and $\boldsymbol{F} \in \mathrm{cwff}_{\alpha\to o}(\Sigma)\big\downarrow_\beta$, $\overline{\nabla}_\forall^\beta$ implies $(\Pi^\alpha \boldsymbol{F}) \in \mathcal{H}$ iff $(\boldsymbol{F}\boldsymbol{A})\big\downarrow_\beta \in \mathcal{H}$ for every $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$. Thus, we have $\upsilon(\Pi^\alpha \boldsymbol{F}) \equiv \mathtt{T}$ iff $\upsilon(\boldsymbol{F} @^\beta \boldsymbol{A}) \equiv \mathtt{T}$ for every $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$.

This verifies $\mathcal{M}_1^{\mathcal{H}} := (\mathrm{cwff}\big\downarrow_\beta, @^\beta, \mathcal{E}^\beta, \upsilon)$ is a Σ-model. Clearly, $\mathcal{M}_1^{\mathcal{H}} \models \mathcal{H}$ since $\upsilon(\boldsymbol{A}) \equiv \mathtt{T}$ for every $\boldsymbol{A} \in \mathcal{H}$ by definition.

By Theorem 3.62, we have a congruence relation $\dot{\sim}$ on $\mathcal{M}_1^{\mathcal{H}}$ induced by Leibniz equality. Note that by Lemma 3.61 in the term model $\mathcal{M}_1^{\mathcal{H}}$, for every type α and every $\boldsymbol{A}, \boldsymbol{B} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$, we have $\boldsymbol{A}_\alpha \dot{\sim} \boldsymbol{B}_\alpha$, iff $\upsilon(\boldsymbol{A} \doteq \boldsymbol{B}) \equiv \mathtt{T}$, iff $(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \mathscr{H}$. Furthermore, if primitive equality is in the signature, then $\mathscr{H} \in \mathfrak{Hint}_*$ is a Hintikka set with primitive equality. Hence, $\mathscr{H}$ satisfies $\overline{\nabla}_=^{\doteq}$ by Lemma 6.22. We have $\boldsymbol{A} \dot{\sim} \boldsymbol{B}$, iff $(\boldsymbol{A} \doteq^\alpha \boldsymbol{B}) \in \mathscr{H}$, iff (by $\overline{\nabla}_=^{\doteq}$) $(\boldsymbol{A} =^\alpha \boldsymbol{B}) \in \mathscr{H}$, iff $\upsilon(\mathscr{E}^\beta(=^\alpha)@^\beta \boldsymbol{A}@^\beta \boldsymbol{B}) \equiv \mathtt{T}$.

Let $\mathscr{M} := \mathscr{M}_1^{\mathscr{H}}/_{\dot{\sim}}$. Each domain of this model has cardinality at most $\aleph_s$ as it is the quotient of a set of cardinality $\aleph_s$. By Theorem 3.62, we know the quotient model $\mathscr{M}$ models $\mathscr{H}$, satisfies property $\mathfrak{q}$, and is a model with primitive equality (if primitive equality is in the signature). Hence, $\mathscr{M} \in \mathfrak{M}_\beta$. Now, we can use Lemma 3.58 to check $\mathscr{M} \in \mathfrak{M}_*$ by checking certain properties of $\dot{\sim}$.

When $* \in \{\beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$, we must check that $\dot{\sim}$ has only two equivalence classes in $\mathscr{D}_o^\beta$. To show this, first note that $\overline{\nabla}_\mathfrak{b}$ holds for $\mathscr{H}$ by Lemma 6.26. Choose any β-normal $\boldsymbol{B} \in \mathscr{H}$. By $\vec{\nabla}_c$, $\neg\boldsymbol{B} \notin \mathscr{H}$. By $\overline{\nabla}_\mathfrak{b}$, for every $\boldsymbol{A} \in \mathrm{cwff}_o(\Sigma)\big\downarrow_\beta$ either $(\boldsymbol{A} \doteq^o \boldsymbol{B})$ or $(\boldsymbol{A} \doteq^o \neg\boldsymbol{B})$. That is, in $\mathscr{M}_1^{\mathscr{H}}$, for every $\boldsymbol{A} \in \mathrm{cwff}_o(\Sigma)\big\downarrow_\beta$ we either have $\boldsymbol{A} \dot{\sim} \boldsymbol{B}$ or $\boldsymbol{A} \dot{\sim} \neg\boldsymbol{B}$. So, we know $\mathscr{M}$ satisfies property $\mathfrak{b}$.

When $* \in \{\beta\eta, \beta\eta\mathfrak{b}\}$, the fact that $\dot{\sim}$ satisfies property η follows from $\overline{\nabla}_\eta$ which holds for $\mathscr{H}$ by Lemma 6.27.

When $* \in \{\beta\xi, \beta\xi\mathfrak{b}\}$, we must show that $\dot{\sim}$ satisfies property ξ. Let $\boldsymbol{M}, \boldsymbol{N} \in \mathrm{wff}_\beta(\Sigma)$, an assignment φ and a variable X_α be given. Suppose $\mathscr{E}^\beta_{\varphi,[\boldsymbol{A}/X]}(\boldsymbol{M}) \dot{\sim} \mathscr{E}^\beta_{\varphi,[\boldsymbol{A}/X]}(\boldsymbol{N})$ for every $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$. Let θ be the substitution defined by $\theta(Y) := \varphi(Y)$ for each variable $Y \in (\mathrm{free}(\boldsymbol{M}) \cup \mathrm{free}(\boldsymbol{N})) \setminus \{X\}$. So, for each $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$,

$$([\boldsymbol{A}/X]\theta(\boldsymbol{M}))\big\downarrow_\beta \equiv \mathscr{E}^\beta_{\varphi,[\boldsymbol{A}/X]}(\boldsymbol{M}) \dot{\sim} \mathscr{E}^\beta_{\varphi,[\boldsymbol{A}/X]}(\boldsymbol{N}) \equiv ([\boldsymbol{A}/X]\theta(\boldsymbol{N}))\big\downarrow_\beta.$$

That is, $([\boldsymbol{A}/X]\theta(\boldsymbol{M}) \doteq^\beta [\boldsymbol{A}/X]\theta(\boldsymbol{N}))\big\downarrow_\beta \in \mathscr{H}$ for every $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$. By $\overline{\nabla}_\xi^\beta$ (Lemma 6.28), we have $((\lambda X.\theta(\boldsymbol{M})) \doteq^{\alpha\to\beta} \lambda X.\theta(\boldsymbol{N}))\big\downarrow_\beta \in \mathscr{H}$. So,

$$\mathscr{E}^\beta_\varphi(\lambda X.\boldsymbol{M}) \equiv (\lambda X.\theta(\boldsymbol{M}))\big\downarrow_\beta \dot{\sim} (\lambda X.\theta(\boldsymbol{N}))\big\downarrow_\beta \equiv \mathscr{E}^\beta_\varphi(\lambda X.\boldsymbol{N}).$$

Thus, $\dot{\sim}$ satisfies ξ as desired.

When $* \in \{\beta\mathfrak{f}, \beta\mathfrak{f}\mathfrak{b}\}$, we must show $\dot{\sim}$ is functional. Let α and β be types and $\boldsymbol{G}, \boldsymbol{H} \in \mathrm{cwff}_{\alpha\to\beta}(\Sigma)\big\downarrow_\beta$. We need to show $\boldsymbol{G} \dot{\sim} \boldsymbol{H}$ iff $(\boldsymbol{G}\boldsymbol{A})\big\downarrow_\beta \dot{\sim} (\boldsymbol{H}\boldsymbol{A})\big\downarrow_\beta$ for every $\boldsymbol{A} \in \mathrm{cwff}_\alpha(\Sigma)\big\downarrow_\beta$. This follows directly from $\overline{\nabla}_\mathfrak{f}^\beta$.

This verifies the fact that $\mathscr{M} \in \mathfrak{M}_*$ whenever $\mathscr{H} \in \mathfrak{Hint}_*$. ⊣

Theorem 6.34 (Model existence theorem). *Let Γ_Σ be a saturated abstract consistency class and let $\Phi \in \Gamma_\Sigma$ be a sufficiently Σ-pure set of sentences. For all $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$ we have: If Γ_Σ is an $\mathfrak{Acc}_*$ (cf. Definition 6.7), then there exists a model $\mathscr{M} \in \mathfrak{M}_*$ (cf. Definition 3.49) that satisfies Φ. Furthermore, each domain of $\mathscr{M}$ has cardinality at most $\aleph_s$.*

Proof. Let Γ_Σ be an abstract consistency class. We can assume without loss of generality (cf. Lemma 6.18) that Γ_Σ is compact, so the preconditions of Lemma 6.32 are met. Therefore, there exists a saturated Hintikka set $\mathscr{H} \in \mathfrak{Hint}_*$ with $\Phi \subseteq \mathscr{H}$. The proof is completed by a simple appeal to the Theorem 6.33. ⊣

THEOREM 6.35 (Model existence for Henkin models). *Let Γ_Σ be a saturated abstract consistency class in $\mathfrak{Acc}_{\beta\mathfrak{fb}}$ and let $\Phi \in \Gamma_\Sigma$ be a sufficiently Σ-pure set of sentences. Then there is a Henkin model (cf. Definition 3.50) that satisfies Φ. Furthermore, each domain of the model has cardinality at most $\aleph_s$.*

PROOF. By Theorem 6.34, there is a model $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{fb}}$ with $\mathcal{M} \models \Phi$. By Theorem 3.68, there is a Henkin model $\mathcal{M}^{fr} \in \mathfrak{M}_{\beta\mathfrak{fb}}$ isomorphic to $\mathcal{M}$. By the isomorphism, we have $\mathcal{M}^{fr} \models \Phi$ and that each domain of $\mathcal{M}^{fr}$ has the same cardinality as the corresponding domain of $\mathcal{M}$. ⊣

REMARK 6.36. The model existence theorems show there are "enough" models in each class $\mathfrak{M}_*$ to model sufficiently pure sets in saturated abstract consistency classes in $\mathfrak{Acc}_*$. These results are abstract forms of completeness. To complete the analysis, we can show abstract forms of soundness. One way to show this is to define a class of sentences

$$\Gamma_\Sigma^* := \{\, \Phi \subseteq \mathrm{cwff}_o \mid \exists \mathcal{M} \in \mathfrak{M}_*.\mathcal{M} \models \Phi \,\}$$

for each $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{fb}\}$ and show Γ_Σ^* is a (saturated) $\mathfrak{Acc}_*$. We only sketch the proof here.

The fact that each Γ_Σ^* satisfy ∇_c, ∇_β, $\nabla_\neg$, $\nabla_\vee$, $\nabla_\wedge$, $\nabla_\forall$, and ∇_{sat} is straightforward. The proof that $\nabla_\exists$ holds has the technical difficulty that one must modify the evaluation of a parameter. Showing $\nabla_\mathfrak{b}$ $[\nabla_\eta]$ holds when considering models with property $\mathfrak{b}$ $[\eta]$ is also easy.

When showing $\nabla_\mathfrak{f}$ holds in $\Gamma_\Sigma^{\beta\mathfrak{f}}$ $[\Gamma_\Sigma^{\beta\mathfrak{fb}}]$, one sees the importance of assuming property q holds. Suppose $\Phi \in \Gamma_\Sigma^{\beta\mathfrak{f}}$ $[\Gamma_\Sigma^{\beta\mathfrak{fb}}]$ and $\neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G}) \in \Phi$. Then there is a model $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_{\beta\mathfrak{f}}$ $[\mathfrak{M}_{\beta\mathfrak{fb}}]$ such that $\mathcal{M} \models \Phi$. This implies $\mathcal{M} \models \neg(\boldsymbol{F} \doteq^{\alpha\to\beta} \boldsymbol{G})$. Without using property q, it follows by Lemma 4.2(1) that $\mathcal{E}(\boldsymbol{F}) \not\equiv \mathcal{E}(\boldsymbol{G})$. By functionality, there is an $\mathsf{a} \in \mathcal{D}_\alpha$ such that $\mathcal{E}(\boldsymbol{F})@\mathsf{a} \not\equiv \mathcal{E}(\boldsymbol{G})@\mathsf{a}$. Let φ be any assignment into $\mathcal{M}$. Then $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{F}X) \not\equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{G}X)$. Now, using property q, we can conclude $\mathcal{M}_{\varphi,[\mathsf{a}/X]} \models \neg((\boldsymbol{F}X) \doteq^\beta (\boldsymbol{G}X))$ by Lemma 4.2(2). Let $w_\alpha \in \Sigma$ be a parameter that does not occur in any sentence of Φ. With some technical work which we omit, one can change the evaluation function to $\mathcal{E}'$ so that $\mathcal{E}'(\boldsymbol{A}) \equiv \mathcal{E}(\boldsymbol{A})$ for all $\boldsymbol{A} \in \Phi$, and $\mathcal{E}'(w) \equiv \mathsf{a}$. In the new model $\mathcal{M}' \equiv (\mathcal{D}, @, \mathcal{E}', \upsilon)$, we have $\mathcal{M}' \models \Phi$ and $\mathcal{M}' \models \neg(\boldsymbol{F}w \doteq^\beta \boldsymbol{G}w)$. Also, $\mathcal{M}' \in \mathfrak{Acc}_{\beta\mathfrak{f}}$ $[\mathfrak{Acc}_{\beta\mathfrak{fb}}]$. This shows $\Phi * \neg(\boldsymbol{F}w \doteq^\beta \boldsymbol{G}w) \in \Gamma_\Sigma^{\beta\mathfrak{f}}$ $[\Gamma_\Sigma^{\beta\mathfrak{fb}}]$. The proof that ∇_ξ holds in $\Gamma_\Sigma^{\beta\xi}$ $[\Gamma_\Sigma^{\beta\xi\mathfrak{b}}]$ is analogous.

We have now established a set of proof-theoretic conditions that are sufficient to guarantee the existence of a model.

§7. Characterizing higher-order natural deduction calculi.

In this section we apply the model existence theorems above to prove some classical higher-order calculi of natural deduction sound and complete with respect to the model classes introduced in Section 3. The first calculus for such a formulation of higher-order logic was a Hilbert-style system introduced by Alonzo Church in [18][10]. Leon Henkin proves completeness (with respect to Henkin models) for a similar calculus with full extensionality in [26]. Peter Andrews introduced a weaker calculus $\mathfrak{T}_\beta$ [1], which lacks all

[10]Church included functional extensionality axioms but only mentions the Boolean extensionality axiom as an option.

$$\frac{\boldsymbol{A} \in \Phi}{\Phi \Vdash \boldsymbol{A}}\,\mathfrak{NR}(Hyp) \qquad \frac{\boldsymbol{A} \equiv_\beta \boldsymbol{B} \quad \Phi \Vdash \boldsymbol{A}}{\Phi \Vdash \boldsymbol{B}}\,\mathfrak{NR}(\beta)$$

$$\frac{\Phi * \boldsymbol{A} \Vdash \boldsymbol{F}_o}{\Phi \Vdash \neg \boldsymbol{A}}\,\mathfrak{NR}(\neg I) \qquad \frac{\Phi \Vdash \neg \boldsymbol{A} \quad \Phi \Vdash \boldsymbol{A}}{\Phi \Vdash \boldsymbol{C}}\,\mathfrak{NR}(\neg E)$$

$$\frac{\Phi \Vdash \boldsymbol{A}}{\Phi \Vdash \boldsymbol{A} \vee \boldsymbol{B}}\,\mathfrak{NR}(\vee I_L) \qquad \frac{\Phi \Vdash \boldsymbol{B}}{\Phi \Vdash \boldsymbol{A} \vee \boldsymbol{B}}\,\mathfrak{NR}(\vee I_R)$$

$$\frac{\Phi \Vdash \boldsymbol{A} \vee \boldsymbol{B} \quad \Phi * \boldsymbol{A} \Vdash \boldsymbol{C} \quad \Phi * \boldsymbol{B} \Vdash \boldsymbol{C}}{\Phi \Vdash \boldsymbol{C}}\,\mathfrak{NR}(\vee E)$$

$$\frac{\Phi \Vdash \boldsymbol{G} w_\alpha \quad w \text{ parameter not occurring in } \Phi \text{ or } \boldsymbol{G}}{\Phi \Vdash \Pi^\alpha \boldsymbol{G}}\,\mathfrak{NR}(\Pi I)^w$$

$$\frac{\Phi \Vdash \Pi^\alpha \boldsymbol{G}}{\Phi \Vdash \boldsymbol{G}\boldsymbol{A}}\,\mathfrak{NR}(\Pi E) \qquad \frac{\Phi * \neg \boldsymbol{A} \Vdash \boldsymbol{F}_o}{\Phi \Vdash \boldsymbol{A}}\,\mathfrak{NR}(Contr)$$

FIGURE 6. Inference rules for $\mathfrak{NR}_\beta$.

forms of extensionality. This calculus has been widely used as a syntactic measure of completeness for machine-oriented calculi [1, 32, 33, 34, 42, 36, 37].

Instead of applying our methods to Hilbert-style calculi, we will use a collection of natural deduction calculi to avoid the tedious details of proving a deduction theorem and propositional completeness. Moreover, natural deduction calculi are more relevant in practice. They form the logical basis for semi-automated theorem proving systems such as HOL [25], ISABELLE [46], or ΩMEGA [51].

DEFINITION 7.1 (The calculi $\mathfrak{NR}_*$). The calculus $\mathfrak{NR}_\beta$ consists of the inference rules[11] in Figure 6 for the provability judgment $\Vdash$ between sets of sentences Φ and sentences $\boldsymbol{A}$. (We write $\Vdash \boldsymbol{A}$ for $\emptyset \Vdash \boldsymbol{A}$.) The rule $\mathfrak{NR}(\beta)$ incorporates β-equality into $\Vdash$. The others characterize the semantics of the connectives and quantifiers.

For $* \in \{\beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$ we obtain the calculus $\mathfrak{NR}_*$ by adding the rules shown in Figure 7 when specified in $*$.

REMARK 7.2. It is worth noting that there is a derivation of $\Vdash \boldsymbol{T}_o$ (i.e., $\Vdash \forall P_0 . P \vee \neg P$) which only uses the rules in Figure 6. Let p be a parameter of type o. A derivation of $\neg(p \vee \neg p) \Vdash (p \vee \neg p)$ is shown in Figure 8. Using $\mathfrak{NR}(Hyp)$ and

[11] Recall that $\boldsymbol{F}_o$ is defined to be $\neg(\forall P_o . (P \vee \neg P))$ and $\mathcal{M} \not\models \boldsymbol{F}_o$ for each Σ-model $\mathcal{M}$ (cf. Lemma 3.43).

$$\frac{A \equiv_{\beta\eta} B \quad \Phi \Vdash A}{\Phi \Vdash B}\,\mathfrak{NK}(\eta) \qquad \frac{\Phi \Vdash \forall X_\alpha . M \doteq^{\beta} N}{\Phi \Vdash (\lambda X_\alpha . M) \doteq^{\alpha\to\beta} (\lambda X_\alpha . N)}\,\mathfrak{NK}(\xi)$$

$$\frac{\Phi \Vdash \forall X_\alpha . GX \doteq^{\beta} HX}{\Phi \Vdash G \doteq^{\alpha\to\beta} H}\,\mathfrak{NK}(\mathfrak{f})$$

$$\frac{\Phi * A \Vdash B \quad \Phi * B \Vdash A}{\Phi \Vdash A \doteq^{o} B}\,\mathfrak{NK}(\mathfrak{b})$$

FIGURE 7. Extensional inference rules.

$$\dfrac{\dfrac{}{\neg(p \vee \neg p), p \Vdash \neg(p \vee \neg p)}\,\mathfrak{NK}(Hyp) \qquad \dfrac{\dfrac{}{\neg(p \vee \neg p), p \Vdash p}\,\mathfrak{NK}(Hyp)}{\neg(p \vee \neg p), p \Vdash (p \vee \neg p)}\,\mathfrak{NK}(\vee I_L)}{\dfrac{\neg(p \vee \neg p), p \Vdash F_o}{\dfrac{\neg(p \vee \neg p) \Vdash \neg p}{\neg(p \vee \neg p) \Vdash (p \vee \neg p)}\,\mathfrak{NK}(\vee I_R)}\,\mathfrak{NK}(\neg I)}\,\mathfrak{NK}(\neg E)$$

FIGURE 8. Derivation of $\neg(p \vee \neg p) \Vdash (p \vee \neg p)$.

$\mathfrak{NK}(\neg E)$, we obtain $\neg(p \vee \neg p) \Vdash F_o$. So, we can conclude $\Vdash (p \vee \neg p)$ using $\mathfrak{NK}(Contr)$. Finally, we obtain a derivation of $\Vdash T_o$ using $\mathfrak{NK}(\Pi I)^p$. Hence, $\Vdash T_o$ is derivable in each calculus $\mathfrak{NK}_*$ where $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. Also, we can apply the rule $\mathfrak{NK}(\Pi E)$ to the end of this derivation with any sentence A to derive $\Vdash (A \vee \neg A)$.

Note that $\mathfrak{NK}_\beta$ and $\mathfrak{NK}_{\beta\mathfrak{f}\mathfrak{b}}$ correspond to the extremes of the model classes discussed in Section 3 (cf. Figure 1 in the introduction). Standard models do not admit (recursively axiomatizable) calculi that are sound and complete, $\mathfrak{NK}_{\beta\mathfrak{f}\mathfrak{b}}$ is complete for Henkin models, and $\mathfrak{NK}_\beta$ is complete for $\mathfrak{M}_\beta$. We will now show soundness and completeness of each $\mathfrak{NK}_*$ with respect to each corresponding model class $\mathfrak{M}_*$ by using the model existence theorems in Section 6.

THEOREM 7.3 (Soundness). *$\mathfrak{NK}_*$ is sound for $\mathfrak{M}_*$ for $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. That is, if $\Phi \Vdash_{\mathfrak{NK}_*} C$ is derivable, then $\mathcal{M} \models C$ for all models $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon)$ in $\mathfrak{M}_*$ such that $\mathcal{M} \models \Phi$.*

PROOF. This can be shown by a simple induction on the derivation of $\Phi \Vdash_{\mathfrak{NK}_*} C$. We distinguish based on the last rule of the derivation. The only base case is $\mathfrak{NK}(Hyp)$, which is trivial since $\mathcal{M} \models C$ whenever $\mathcal{M} \models \Phi$ and $C \in \Phi$.

$\mathfrak{NR}(\beta)$: Suppose $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash \boldsymbol{A}$ and $\boldsymbol{A} \equiv_\beta \boldsymbol{C}$. Let $\mathcal{M} \in \mathfrak{M}_*$ be a model of Φ. By induction, we know $\mathcal{M} \models \boldsymbol{A}$ and so $\mathcal{M} \models \boldsymbol{C}$ using Remark 3.19.

$\mathfrak{NR}(Contr)$: Suppose $\mathcal{M} \in \mathfrak{M}_*$, $\mathcal{M} \models \Phi$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi * \neg \boldsymbol{C} \Vdash \boldsymbol{F}_o$. By Lemma 3.43, $\mathcal{M} \not\models \boldsymbol{F}_o$. So, we must have $\mathcal{M} \not\models \neg \boldsymbol{C}$. Hence, $\mathcal{M} \models \boldsymbol{C}$.

$\mathfrak{NR}(\neg I)$: Analogous to $\mathfrak{NR}(Contr)$.

$\mathfrak{NR}(\neg E)$: Suppose $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash \neg \boldsymbol{A}$ and $\Phi \Vdash \boldsymbol{A}$. By induction, any model in $\mathfrak{M}_*$ of Φ would have to model both $\boldsymbol{A}$ and $\neg \boldsymbol{A}$. So, there is no such model of Φ and we are done.

$\mathfrak{NR}(\vee I_L)$: Suppose $\mathcal{M} \in \mathfrak{M}_*$, $\mathcal{M} \models \Phi$, $\boldsymbol{C}$ is $(\boldsymbol{A} \vee \boldsymbol{B})$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash \boldsymbol{A}$. By induction, $\mathcal{M} \models \boldsymbol{A}$ and so $\mathcal{M} \models (\boldsymbol{A} \vee \boldsymbol{B})$.

$\mathfrak{NR}(\vee I_R)$: Analogous to $\mathfrak{NR}(\vee I_L)$.

$\mathfrak{NR}(\vee E)$: Suppose $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash (\boldsymbol{A} \vee \boldsymbol{B})$, $\Phi * \boldsymbol{A} \Vdash \boldsymbol{C}$ and $\Phi * \boldsymbol{B} \Vdash \boldsymbol{C}$. Let $\mathcal{M} \in \mathfrak{M}_*$ be a model of Φ. By induction, $\mathcal{M} \models \boldsymbol{A} \vee \boldsymbol{B}$. If $\mathcal{M} \models \boldsymbol{A}$, then by induction $\mathcal{M} \models \boldsymbol{C}$ since $\Phi * \boldsymbol{A} \Vdash \boldsymbol{C}$. If $\mathcal{M} \models \boldsymbol{B}$, then by induction $\mathcal{M} \models \boldsymbol{C}$ since $\Phi * \boldsymbol{B} \Vdash \boldsymbol{C}$. In either case, $\Phi \Vdash \boldsymbol{C}$.

$\mathfrak{NR}(\Pi I)$: Suppose $\boldsymbol{C}$ is $(\Pi^\alpha \boldsymbol{G})$ and $\Phi \Vdash (\Pi^\alpha \boldsymbol{G})$ follows from $\Phi \Vdash \boldsymbol{G}w$ where w_α is a parameter which does not occur in any sentence of Φ or in $\boldsymbol{G}$. Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_*$ be a model of Φ. Assume $\mathcal{M} \not\models \Pi^\alpha \boldsymbol{G}$. Then there must be some $\mathsf{a} \in \mathcal{D}_\alpha$ such that $\upsilon(\mathcal{E}(\boldsymbol{G})@\mathsf{a}) \equiv \mathtt{F}$. From the evaluation function $\mathcal{E}$, one can define another evaluation function $\mathcal{E}'$ such that $\mathcal{E}'(w) \equiv \mathsf{a}$ and $\mathcal{E}'_\varphi(\boldsymbol{A}_\alpha) \equiv \mathcal{E}_\varphi(\boldsymbol{A}_\alpha)$ if w does not occur in $\boldsymbol{A}$. Let $\mathcal{M}' := (\mathcal{D}, @, \mathcal{E}', \upsilon)$. One can check $\mathcal{M}' \in \mathfrak{M}_*$ using the fact that $\mathcal{M} \in \mathfrak{M}_*$. Since $\mathcal{M}' \models \Phi$, by induction we have $\mathcal{M}' \models \boldsymbol{G}w$. This contradicts $\upsilon(\mathcal{E}'(\boldsymbol{G})@\mathsf{a}) \equiv \upsilon(\mathcal{E}(\boldsymbol{G})@\mathsf{a}) \equiv \mathtt{F}$. Thus, $\mathcal{M} \models \Pi^\alpha \boldsymbol{G}$.

$\mathfrak{NR}(\Pi E)$: Suppose $\boldsymbol{C}$ is $(\boldsymbol{G}\boldsymbol{A})$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash (\Pi^\alpha \boldsymbol{G})$. Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_*$ be a model of Φ. By induction, $\mathcal{M} \models (\Pi^\alpha \boldsymbol{G})$ and thus $\upsilon(\mathcal{E}(\boldsymbol{G}))@\mathsf{a} \equiv \mathtt{T}$ for every $\mathsf{a} \in \mathcal{D}_\alpha$. In particular, $\mathcal{M} \models \boldsymbol{G}\boldsymbol{A}$.

We now check soundness of the rules in Figure 7 with respect to their model classes:

$\mathfrak{NR}(\eta)$: Analogous to $\mathfrak{NR}(\beta)$ using property η.

$\mathfrak{NR}(\xi)$: Suppose $\boldsymbol{C}$ is $(\lambda X_\alpha \boldsymbol{.} \boldsymbol{M}) \doteq^{\alpha \to \beta} (\lambda X_\alpha \boldsymbol{.} \boldsymbol{N})$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash \forall X_\alpha \boldsymbol{.} \boldsymbol{M} \doteq^\beta \boldsymbol{N}$. Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_*$ be a model of Φ. By induction, we have $\mathcal{M} \models \forall X_\alpha \boldsymbol{.} \boldsymbol{M} \doteq^\beta \boldsymbol{N}$. So, for any assignment φ and $\mathsf{a} \in \mathcal{D}_\alpha$, $\mathcal{M} \models_{\varphi,[\mathsf{a}/X]} \boldsymbol{M} \doteq^\beta \boldsymbol{N}$. Note that property $\mathfrak{q}$ holds in $\mathcal{M}$ since $\mathcal{M} \in \mathfrak{M}_*$ (cf. Definition 3.49). By Lemma 4.2(2), $\mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{M}) \equiv \mathcal{E}_{\varphi,[\mathsf{a}/X]}(\boldsymbol{N})$. By property ξ, $\mathcal{E}_\varphi(\lambda X_\alpha \boldsymbol{.} \boldsymbol{M}) \equiv \mathcal{E}_\varphi(\lambda X_\alpha \boldsymbol{.} \boldsymbol{N})$ and thus $\mathcal{M} \models \boldsymbol{C}$ by Lemma 4.2(1).

$\mathfrak{NR}(\mathfrak{f})$: Suppose $\boldsymbol{C}$ is $\boldsymbol{G} \doteq^{\alpha \to \beta} \boldsymbol{H}$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi \Vdash \forall X_\alpha \boldsymbol{.} \boldsymbol{G}X \doteq^\beta \boldsymbol{H}X$. Let $\mathcal{M} \in \mathfrak{M}_*$ be a model of Φ. By induction, we know $\mathcal{M} \models \forall X_\alpha \boldsymbol{.} \boldsymbol{G}X \doteq^\beta \boldsymbol{H}X$. Note that property $\mathfrak{q}$ holds for $\mathcal{M}$ since $\mathcal{M} \in \mathfrak{M}_*$. By Theorem 4.3(3), we must have $\mathcal{M} \models (\boldsymbol{G} \doteq^{\alpha \to \beta} \boldsymbol{H})$.

$\mathfrak{NR}(\mathfrak{b})$ Suppose $\boldsymbol{C}$ is $\boldsymbol{A} \doteq^o \boldsymbol{B}$ and $\Phi \Vdash \boldsymbol{C}$ follows from $\Phi * \boldsymbol{A} \Vdash \boldsymbol{B}$ and $\Phi * \boldsymbol{B} \Vdash \boldsymbol{A}$. Let $\mathcal{M} \equiv (\mathcal{D}, @, \mathcal{E}, \upsilon) \in \mathfrak{M}_*$ be a model of Φ. If $\mathcal{M} \models \boldsymbol{A}$, then $\mathcal{M} \models \boldsymbol{B}$ by induction. If $\mathcal{M} \models \boldsymbol{B}$, then $\mathcal{M} \models \boldsymbol{A}$ by induction. These facts imply $\upsilon(\mathcal{E}(\boldsymbol{A})) \equiv \upsilon(\mathcal{E}(\boldsymbol{B}))$. By Lemma 3.48, we have $\mathcal{M} \models (\boldsymbol{A} \Leftrightarrow \boldsymbol{B})$. By Theorem 4.3(4), we must have $\mathcal{M} \models (\boldsymbol{A} \doteq^o \boldsymbol{B})$. $\dashv$

Definition 7.4 ($\mathfrak{NK}_*$-consistent). A set of sentences Φ is $\mathfrak{NK}_*$-*inconsistent* if $\Phi \Vdash_{\mathfrak{NK}_*} \boldsymbol{F}_o$, and $\mathfrak{NK}_*$-*consistent* otherwise.

Now, we use the model existence theorems for $\mathscr{HOL}$ to give short and elegant proofs of completeness for $\mathfrak{NK}_*$.

Lemma 7.5. *The class* $\Gamma_\Sigma^* := \{\Phi \subseteq \mathrm{cwff}_o \mid \Phi$ *is* $\mathfrak{NK}_*$-*consistent*$\}$ *is a saturated* $\mathfrak{Acc}_*$.

Proof. Obviously Γ_Σ^* is closed under subsets, since any subset of an $\mathfrak{NK}_*$-consistent set is $\mathfrak{NK}_*$-consistent. We now check the remaining conditions. We prove all the properties by proving their contrapositive.

∇_c: Suppose $\boldsymbol{A}, \neg\boldsymbol{A} \in \Phi$. We have $\Phi \Vdash \boldsymbol{F}_o$ by $\mathfrak{NK}(Hyp)$ and $\mathfrak{NK}(\neg E)$.

∇_β: Let $\boldsymbol{A} \in \Phi$, $\boldsymbol{A} \equiv_\beta \boldsymbol{B}$ and $\Phi * \boldsymbol{B}$ be $\mathfrak{NK}_*$-inconsistent. That is, $\Phi * \boldsymbol{B} \Vdash \boldsymbol{F}_o$. By $\mathfrak{NK}(\neg I)$, we know $\Phi \Vdash \neg\boldsymbol{B}$. Since $\boldsymbol{A} \in \Phi$, we know $\Phi \Vdash \boldsymbol{B}$ by $\mathfrak{NK}(Hyp)$ and $\mathfrak{NK}(\beta)$. Using $\mathfrak{NK}(\neg E)$, we know $\Phi \Vdash \boldsymbol{F}_o$ and hence Φ is $\mathfrak{NK}_*$-inconsistent.

$\nabla_\neg$: Suppose $\neg\neg\boldsymbol{A} \in \Phi$ and $\Phi * \boldsymbol{A}$ is $\mathfrak{NK}_*$-inconsistent. From $\Phi * \boldsymbol{A} \Vdash \boldsymbol{F}_o$ and $\mathfrak{NK}(\neg I)$, we have $\Phi \Vdash \neg\boldsymbol{A}$. Since $\neg\neg\boldsymbol{A} \in \Phi$, we can apply $\mathfrak{NK}(Hyp)$ and $\mathfrak{NK}(\neg E)$ to obtain $\Phi \Vdash \boldsymbol{F}_o$.

$\nabla_\vee$: Suppose $(\boldsymbol{A} \vee \boldsymbol{B}) \in \Phi$ and both $\Phi * \boldsymbol{A}$ and $\Phi * \boldsymbol{B}$ are $\mathfrak{NK}_*$-inconsistent. By $\mathfrak{NK}(Hyp)$ and $\mathfrak{NK}(\vee E)$, we have $\Phi \Vdash \boldsymbol{F}_o$.

$\nabla_\wedge$: Suppose $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \Phi$ and $\Phi * \neg\boldsymbol{A} * \neg\boldsymbol{B}$ is $\mathfrak{NK}_*$-inconsistent. By $\mathfrak{NK}(Contr)$ and $\mathfrak{NK}(\vee I_R)$, we have $\Phi, \neg\boldsymbol{A} \Vdash \boldsymbol{A} \vee \boldsymbol{B}$. Using $\mathfrak{NK}(\neg E)$ with $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \Phi$, we have $\Phi, \neg\boldsymbol{A} \Vdash \boldsymbol{F}_o$. By $\mathfrak{NK}(Contr)$ and $\mathfrak{NK}(\vee I_L)$, we have $\Phi \Vdash \boldsymbol{A} \vee \boldsymbol{B}$. Using $\mathfrak{NK}(\neg E)$ with $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \in \Phi$, Φ is $\mathfrak{NK}_*$-inconsistent.

$\nabla_\forall$: Suppose $(\Pi^\alpha \boldsymbol{G}) \in \Phi$ and $\Phi * (\boldsymbol{GA})$ is $\mathfrak{NK}_*$-inconsistent. By $\mathfrak{NK}(\neg I)$, $\Phi \Vdash \neg(\boldsymbol{GA})$. By $\mathfrak{NK}(Hyp)$ and $\mathfrak{NK}(\Pi E)$, $\Phi \Vdash \boldsymbol{GA}$. Finally, $\mathfrak{NK}(\neg E)$ implies $\Phi \Vdash \boldsymbol{F}_o$.

$\nabla_\exists$: Suppose $\neg(\Pi^\alpha \boldsymbol{G}) \in \Phi$, w_α is a parameter which does not occur in Φ, and $\Phi * \neg(\boldsymbol{G}w)$ is $\mathfrak{NK}_*$-inconsistent. By $\mathfrak{NK}(Contr)$, $\Phi \Vdash \boldsymbol{G}w$. By $\mathfrak{NK}(\Pi I)^w$, $\Phi \Vdash (\Pi^\alpha \boldsymbol{G})$. Using $\mathfrak{NK}(\neg E)$ with $\neg(\Pi^\alpha \boldsymbol{G}) \in \Phi$, Φ is $\mathfrak{NK}_*$-inconsistent.

∇_{sat}: Let $\Phi * \boldsymbol{A}$ and $\Phi * \neg\boldsymbol{A}$ be $\mathfrak{NK}_*$-inconsistent. We show that Φ is $\mathfrak{NK}_*$-inconsistent. Using $\mathfrak{NK}(\neg I)$, we know $\Phi \Vdash \neg\boldsymbol{A}$ and $\Phi \Vdash \neg\neg\boldsymbol{A}$. By $\mathfrak{NK}(\neg E)$, we have $\Phi \Vdash \boldsymbol{F}_o$.

Thus we have shown that Γ_Σ^β is saturated and in $\mathfrak{Acc}_\beta$. Now let us check the conditions for the additional properties η, ξ, $\mathfrak{f}$, and $\mathfrak{b}$.

∇_η: If $*$ includes η, then the proof proceeds as in ∇_β above, but with the rule $\mathfrak{NK}(\eta)$.

∇_ξ: Suppose $*$ includes ξ, $\neg(\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N}) \in \Phi$, and $\Phi * \neg([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})$ is $\mathfrak{NK}_*$-inconsistent for some parameter w_α which does not occur in any sentence of Φ. By $\mathfrak{NK}(Contr)$, we have $\Phi \Vdash ([w/X]\boldsymbol{M} \doteq^\beta [w/X]\boldsymbol{N})$. By $\mathfrak{NK}(\beta)$, we have $\Phi \Vdash ((\lambda X.\boldsymbol{M} \doteq^\beta \boldsymbol{N})w)$. By $\mathfrak{NK}(\Pi I)$, $\Phi \Vdash (\forall X.\boldsymbol{M} \doteq^\beta \boldsymbol{N})$. By $\mathfrak{NK}(\xi)$, $\Phi \Vdash (\lambda X.\boldsymbol{M} \doteq^{\alpha\to\beta} \lambda X.\boldsymbol{N})$. By $\mathfrak{NK}(\neg E)$, Φ is $\mathfrak{NK}_*$-inconsistent.

$\nabla_\mathfrak{f}$: This case is analogous to the previous one, generalizing $\lambda X.\boldsymbol{M} \doteq \lambda X.\boldsymbol{N}$ to arbitrary $\boldsymbol{G} \doteq \boldsymbol{H}$ and using the extensionality rule $\mathfrak{NK}(\mathfrak{f})$ instead of $\mathfrak{NK}(\xi)$.

$\nabla_\mathfrak{b}$: Suppose $*$ includes $\mathfrak{b}$. Assume that $\neg(\boldsymbol{A} \doteq^o \boldsymbol{B}) \in \Phi$ but both $\Phi * \neg\boldsymbol{A} * \boldsymbol{B} \notin \Gamma_\Sigma^*$ and $\Phi * \boldsymbol{A} * \neg\boldsymbol{B} \notin \Gamma_\Sigma^*$. So both are $\mathfrak{NK}_*$-inconsistent and we have $\Phi * \boldsymbol{A} \Vdash \boldsymbol{B}$ and $\Phi * \boldsymbol{B} \Vdash \boldsymbol{A}$ by $\mathfrak{NK}(Contr)$. By $\mathfrak{NK}(\mathfrak{b})$, we have $\Phi \Vdash (\boldsymbol{A} \doteq^o \boldsymbol{B})$. Since $\neg(\boldsymbol{A} \doteq^o \boldsymbol{B}) \in \Phi$, Φ is $\mathfrak{NK}_*$-inconsistent. ⊣

THEOREM 7.6 (Henkin's theorem for $\mathfrak{NK}_*$). *Let* $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. *Every sufficiently* Σ*-pure* $\mathfrak{NK}_*$*-consistent set of sentences has an* $\mathfrak{M}_*$*-model.*

PROOF. Let Φ be a sufficiently Σ-pure $\mathfrak{NK}_*$-consistent set of sentences. By Theorem 7.5 we know that the class of sets of $\mathfrak{NK}_*$-consistent sentences constitute a saturated $\mathfrak{Acc}_*$, thus the Model Existence Theorem (Theorem 6.34) guarantees an $\mathfrak{M}_*$ model for Φ. ⊣

COROLLARY 7.7 (Completeness theorem for $\mathfrak{NK}_*$). *Let* Φ *be a sufficiently* Σ*-pure set of sentences,* $\boldsymbol{A}$ *be a sentence, and* $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. *If* $\boldsymbol{A}$ *is valid in all models* $\mathcal{M} \in \mathfrak{M}_*$ *that satisfy* Φ, *then* $\Phi \Vdash_{\mathfrak{NK}_*} \boldsymbol{A}$.

PROOF. Let $\boldsymbol{A}$ be given such that $\boldsymbol{A}$ is valid in all $\mathfrak{M}_*$ models that satisfy Φ. So, $\Phi * \neg\boldsymbol{A}$ is unsatisfiable in $\mathfrak{M}_*$. Since only finitely many constants occur in $\neg\boldsymbol{A}$, $\Phi * \neg\boldsymbol{A}$ is sufficiently Σ-pure. So, $\Phi * \neg\boldsymbol{A}$ must be $\mathfrak{NK}_*$-inconsistent by Henkin's theorem above. Thus, $\Phi \Vdash_{\mathfrak{NK}_*} \boldsymbol{A}$ by $\mathfrak{NK}(Contr)$. ⊣

Finally we can use the completeness theorems obtained so far to prove a compactness theorem for our semantics.

COROLLARY 7.8 (Compactness theorem for $\mathfrak{NK}_*$). *Let* Φ *be a sufficiently* Σ*-pure set of sentences and* $* \in \{\beta, \beta\eta, \beta\xi, \beta\mathfrak{f}, \beta\mathfrak{b}, \beta\eta\mathfrak{b}, \beta\xi\mathfrak{b}, \beta\mathfrak{f}\mathfrak{b}\}$. Φ *has an* $\mathfrak{M}_*$*-model iff every finite subset of* Φ *has an* $\mathfrak{M}_*$*-model.*

PROOF. If Φ has no $\mathfrak{M}_*$-model, then by Theorem 7.6 Φ is $\mathfrak{NK}_*$-inconsistent. Since every $\mathfrak{NK}_*$-proof is finite, this means some finite subset Ψ of Φ is $\mathfrak{NK}_*$-inconsistent. Hence, Ψ has no $\mathfrak{M}_*$-model. ⊣

REMARK 7.9 (Calculi with primitive equality). If primitive equality is included in the signature, a simple way of extending the calculi $\mathfrak{NK}_*$ in a sound and complete way is to include the rules $\mathfrak{NK}(=^r)$ and $\mathfrak{NK}(=^l)$ in Figure 9. These rules are clearly sound for models with primitive equality. One can argue completeness by showing $\Gamma^*_\Sigma := \{\Phi \subseteq \mathrm{wff}_o(\Sigma) \mid \Phi \text{ is } \mathfrak{NK}_*\text{-consistent}\}$ is a saturated $\mathfrak{Acc}_*$ with primitive equality. By Lemma 7.5, we already know Γ^*_Σ is a saturated $\mathfrak{Acc}_*$. To show the conditions for primitive equality, one can show Γ^*_Σ satisfies $\nabla^r_=$ using $\mathfrak{NK}(=^r)$ and $\nabla^{\dot{=}}_=$ using $\mathfrak{NK}(=^l)$.

$$\frac{}{\Phi \Vdash \boldsymbol{A} =^\alpha \boldsymbol{A}}\ \mathfrak{NK}(=^r) \qquad\qquad \frac{\Phi \Vdash \boldsymbol{C} =^\alpha \boldsymbol{D}}{\Phi \Vdash \boldsymbol{C} \doteq^\alpha \boldsymbol{D}}\ \mathfrak{NK}(=^l)$$

FIGURE 9. Primitive equality in $\mathfrak{NK}_*$.

§8. Conclusion. In this article, we have given an overview of the landscape of semantics for classical higher-order logics. We have differentiated nine different possible notions and have tied the discerning properties to conditions of corresponding abstract consistency classes. The practical relevance of these notions has been illustrated by pointing to application scenarios within mathematics, programming languages, and computational linguistics.

Our model existence theorems are strong proof tools connecting syntax and semantics. A standard application is in completeness analysis of higher-order calculi. A calculus $\mathscr{C}$ is shown to be complete for a model class $\mathfrak{M}_*$ by showing that the class of $\mathscr{C}$-consistent or $\mathscr{C}$-irrefutable sets of sentences is in $\mathfrak{Acc}_*$. Then completeness follows from the model existence results. We have given an example of this by showing completeness for natural deduction calculi in Section 7.

8.1. Applications and related work. The generalized model classes $\mathfrak{M}_*$ have many possible applications. An example is higher-order logic programming [45] where the denotational semantics of programs can induce non-standard meanings for the classical connectives. For instance, given an SLD-like search strategy as in λ-PROLOG [43], conjunction is not commutative any more. Therefore, various authors have proposed model-theoretic semantics where property $\mathfrak{b}$ fails. David Wolfram, for instance, uses Andrews' υ-complexes [58] as a semantics for λ-PROLOG and Gopalan Nadathur uses "labeled structures" for the same purpose in [45]. Mary DeMarco [20] also develops a model theory for intuitionistic type theory and λ-prolog in which property $\mathfrak{b}$ may fail (James Lipton and Mary DeMarco are continuing this work). Till Mossakowski and Lutz Schröder have been studying non-functional Henkin models for a partial λ-calculus in the context of the HasCasl specification language [48, 49]. It is plausible to assume that the results of this article will be useful for further development in this direction. Further relevance of model-theoretic semantics where property $\mathfrak{q}$ fails, however, is not sufficiently investigated yet, but seems a promising line of research.

The article also provides a basis for the investigation of hyper-intensional semantics of natural languages. In fact early versions of this article have already influenced the work of Lappin and Pollard [40]. Hyper-intensional semantics provide theories for logics where Boolean extensionality (and thus the substitutability of equivalents) can fail. Linguistically motivated theories like the ones presented in [56, 17, 41, 40] introduce intensional (non-standard) variants of the connectives and quantifiers acting on a generalized domain of truth values. Interestingly, only [41] and [40] present formal model-theoretic semantics. The model construction in [41] strongly resembles Peter Andrew's υ-complexes (semantic objects are paired with syntactic representations; in this case linguistic parse trees). In [40], $\mathscr{D}_o$ is taken to be a pre-Boolean algebra, and possible worlds are associated with ultrafilters. A direct comparison is aggravated by the fact that Lappin and Pollard's work is situated in a Montague-style intensional (i.e., modal) context. A generalization of our work by techniques from [23] seems the way to go here.

8.2. Relaxing the saturation assumption. Unfortunately, the model existence theorems presented in this article do not support completeness proofs for most higher-order machine-oriented calculi, such as higher-order resolution [33, 13], higher-order paramodulation [11], or tableau-based calculi [5, 37]. This is because we had to assume saturation of abstract consistency classes to prove the model existence theorems. The problem is that machine oriented calculi are typically, in some sense, cut-free. This makes saturation very difficult to show.

For the same reason the results of this article also do not apply to another prominent application of model existence theorems: relatively simple (but non-constructive) cut-elimination theorems. In [1] Peter Andrews applies his "Unifying Principle" to cut-elimination in a cut-free non-extensional sequent calculus, by proving the calculus complete (relative to $\mathfrak{T}_\beta$). He concludes that cut-elimination is valid for this calculus. Again, the saturation condition prevents us from obtaining variants of the extensional cut-elimination theorems in [54, 55] by Andrews' approach using our model existence theorem for Henkin models. In fact one can prove (cf. [12]) that the problem of showing that an abstract consistency class can be extended to a saturated one is equivalent to showing cut elimination for certain sequent or resolution calculi.

To account for the saturation problem we have additionally investigated model existence for the model classes presented in this article using an extension of Peter Andrews' υ-complexes (cf. [12]). The model construction in this technique requires an abstract consistency class to satisfy certain *acceptability* conditions which are much weaker than saturation. (For example, the acceptability conditions can be shown to hold for abstract consistency classes obtained from certain cut-free sequent calculi.) Because this technique is much more complex and subtle than the relatively simple quotients of term evaluations used in this article, we did not include the extended results here. The unsaturated model existence theorems imply that every acceptable abstract consistency class can be extended to a saturated one. Armed with this fact, we can use the model existence theorems presented here to rescue the general completeness and cut elimination results mentioned above. To show, for example, completeness of a higher-order machine-oriented calculus $\mathscr{C}$, we define the class Γ of $\mathscr{C}$-irrefutable sentences and show that it is an acceptable (but unsaturated) abstract consistency class. By the extension result in [12] there is a saturated abstract consistency class $\Gamma' \supseteq \Gamma$. By application of saturated model existence from this article we obtain a suitable model for every (sufficiently Σ-pure) $\Phi \in \Gamma'$ and thus for every (sufficiently Σ-pure) $\Phi \in \Gamma$. This immediately gives us completeness. Hence, the leverage added by this article together with [12] is that we can now extend non-extensional cut-elimination results to extensional cases.

Acknowledgments. The work presented in this paper has been supported by the "Deutsche Forschungsgemeinschaft" (DFG) under Grant SI 372/4 HOTEL, the National Science Foundation under Grant CCR-0097179 and a DFG Heisenberg stipend (Ko-1370/6-1) to the third author. The authors would like to thank Peter Andrews and Frank Pfenning for stimulating discussions and Claus-Peter Wirth and Andrey Paskevich for proof reading. We furthermore thank the referee of this article for his very fruitful comments.

REFERENCES

[1] PETER B. ANDREWS, *Resolution in type theory*, this JOURNAL, vol. 36 (1971), no. 3, pp. 414–432.

[2] ———, *General models and extensionality*, this JOURNAL, vol. 37 (1972), no. 2, pp. 395–397.

[3] ———, *General models descriptions and choice in type theory*, this JOURNAL, vol. 37 (1972), no. 2, pp. 385–394.

[4] ———, letter to Roger Hindley dated January 22, 1973.

[5] ———, *On connections and higher order logic*, ***Journal of Automated Reasoning***, vol. 5 (1989), pp. 257–291.

[6] ———, ***An introduction to mathematical logic and type theory: To truth through proof***, second ed., Kluwer Academic Publishers, 2002.

[7] PETER B. ANDREWS, MATTHEW BISHOP, and CHAD E. BROWN, *TPS: A theorem proving system for type theory*, ***Proceedings of the 17th international conference on automated deduction*** (Pittsburgh, USA) (David McAllester, editor), Lecture Notes in Artifical Intelligence, no. 1831, Springer-Verlag, 2000, pp. 164–169.

[8] PETER B. ANDREWS, MATTHEW BISHOP, SUNIL ISSAR, DAN NESMITH, FRANK PFENNING, and HONGWEI XI, *TPS: A theorem proving system for classical type theory*, ***Journal of Automated Reasoning***, vol. 16 (1996), no. 3, pp. 321–353.

[9] HENK P. BARENDREGT, ***The lambda calculus***, North-Holland, 1984.

[10] CHRISTOPH BENZMÜLLER, *Equality and extensionality in automated higher-order theorem proving*, ***Ph.D. thesis***, Saarland University, 1999.

[11] ———, *Extensional higher-order paramodulation and RUE-resolution*, ***Proceedings of the 16th international Conference on Automated Deduction*** (Trento, Italy) (Harald Ganzinger, editor), Lecture Notes in Artificial Intelligence, vol. 1632, Springer-Verlag, 1999, pp. 399–413.

[12] CHRISTOPH BENZMÜLLER, CHAD E. BROWN, and MICHAEL KOHLHASE, *Semantic techniques for higher-order cut-elimination*, manuscript, `http://www.ags.uni-sb.de/~chris/papers/R19.pdf`, 2002.

[13] CHRISTOPH BENZMÜLLER and MICHAEL KOHLHASE, *Extensional higher order resolution*, in Kirchner and Kirchner [35], pp. 56–72.

[14] ———, *LEO—a higher order theorem prover*, in Kirchner and Kirchner [35], pp. 139–144.

[15] ———, *Model existence for higher-order logic*, ***SEKI-Report SR-97-09***, Saarland University, 1997.

[16] Wolfgang Bibel and Peter Schmitt (editors), ***Automated deduction—a basis for applications***, Kluwer, 1998.

[17] GENNARO CHIERCHIA and RAYMOND TURNER, *Semantics and property theory*, ***Linguistics and Philosophy***, vol. 11 (1988), pp. 261–302.

[18] ALONZO CHURCH, *A formulation of the simple theory of types*, this JOURNAL, vol. 5 (1940), pp. 56–68.

[19] NICOLAAS GOVERT DE BRUIJN, *Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with an application to the Church-Rosser theorem*, ***Indagationes Mathematicae***, vol. 34 (1972), no. 5, pp. 381–392.

[20] MARY DEMARCO, *Intuitionistic semantics for heriditarily harrop logic programming*, ***Ph.D. thesis***, Wesleyan University, 1999.

[21] GILLES DOWEK, THÉRÈSE HARDIN, and CLAUDE KIRCHNER, *HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic*, ***Mathematical Structures in Computer Science***, vol. 11 (2001), no. 1, pp. 1–25.

[22] MELVIN FITTING, ***First-order logic and automated theorem proving***, second ed., Graduate Texts in Computer Science, Springer-Verlag, 1996.

[23] ———, ***Types, tableaus, and Gödel's God***, Kluwer, 2002.

[24] KURT GÖDEL, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, ***Monatshefte der Mathematischen Physik***, vol. 38 (1931), pp. 173–198, English version in [57].

[25] M. J. C. GORDON and T. F. MELHAM, ***Introduction to HOL—a theorem proving environment for higher order logic***, Cambridge University Press, 1993.

[26] LEON HENKIN, *Completeness in the theory of types*, this JOURNAL, vol. 15 (1950), no. 2, pp. 81–91.

[27] ———, *The discovery of my completeness proofs*, ***The Bulletin of Symbolic Logic***, vol. 2 (1996), no. 2, pp. 127–158.

[28] ROGER J. HINDLEY and JONATHAN P. SELDIN, ***Introduction to combinators and lambda-calculs***, Cambridge University Press, Cambridge, 1986.

[29] K. J. J. HINTIKKA, *Form and content in quantification theory*, ***Acta Philosophica Fennica***, vol. 8 (1955), pp. 7–55.

[30] FURIO HONSELL and MARINA LENISA, *Coinductive characterizations of applicative structures*, ***Mathematical Structures in Computer Science***, vol. 9 (1999), pp. 403–435.

[31] FURIO HONSELL and DONALD SANNELLA, *Pre-logical relations*, ***Proceedings of computer science logic*** (CSL '99), Lecture Notes in Computer Science, vol. 1683, Springer-Verlag, 1999, pp. 546–561.

[32] GÉRARD P. HUET, *Constrained resolution: A complete method for higher order logic*, ***Ph. D. thesis***, Case Western Reserve University, 1972.

[33] ———, *A mechanization of type theory*, ***Proceedings of the 3rd international joint conference on artificial intelligence*** (Donald E. Walker and Lewis Norton, editors), 1973, pp. 139–146.

[34] D. C. JENSEN and THOMASZ PIETRZYKOWSKI, *A complete mechanization of* (ω)*-order type theory*, ***Proceedings of the ACM annual conference***, vol. 1, 1972, pp. 82–92.

[35] Claude Kirchner and Hélène Kirchner (editors), ***Proceedings of the 15th Conference on Automated Deduction***, Lecture Notes in Artificial Intelligence, vol. 1421, Springer-Verlag, 1998.

[36] MICHAEL KOHLHASE, *A mechanization of sorted higher-order logic based on the resolution principle*, ***Ph. D. thesis***, Saarland University, 1994.

[37] ———, *Higher-order tableaux*, ***Theorem proving with analytic tableaux and related methods*** (Peter Baumgartner, Reiner Hähnle, and Joachim Posegga, editors), Lecture Notes in Artificial Intelligence, vol. 918, Springer-Verlag, 1995, pp. 294–309.

[38] MICHAEL KOHLHASE and ORTWIN SCHEJA, *Higher-order multi-valued resolution*, ***Journal of Applied Non-Classical Logics***, vol. 9 (1999), no. 4, pp. 155–178.

[39] SHALOM LAPPIN and CARL POLLARD, *Strategies for hyperintensional semantics*, manuscript, King's College, London and Ohio State University, 2000.

[40] ———, *A higher-order fine-grained logic for intensional semantics*, manuscript, 2002.

[41] RICHARD LARSON and GABRIEL SEGAL, ***Knowledge of meaning***, MIT Press, 1995.

[42] DALE MILLER, *Proofs in higher-order logic*, ***Ph. D. thesis***, Carnegie-Mellon University, 1983.

[43] ———, *A logic programming language with lambda-abstraction, function variables, and simple unification*, ***Journal of Logic and Computation***, vol. 4 (1991), no. 1, pp. 497–536.

[44] JOHN C. MITCHELL, ***Foundations for programming languages***, Foundations of Computing, MIT Press, 1996.

[45] GOPALAN NADATHUR and DALE MILLER, *Higher-order logic programming*, ***Technical Report CS-1994-38***, Department of Computer Science, Duke University, 1994.

[46] TOBIAS NIPKOW, LAWRENCE C. PAULSON, and MARKUS WENZEL, ***Isabelle/HOL—a proof assistant for higher-order logic***, Lecture Notes in Computer Science, vol. 2283, Springer-Verlag, 2002.

[47] J. ALAN ROBINSON and ANDREI VORONKOV, ***Handbook of automated reasoning***, MIT Press, 2001.

[48] L. SCHRÖDER and T. MOSSAKOWSKI, *Hascasl: towards integrated specification and development of functional programs*, ***Algebraic methodology and software technology***, Lecture Notes in Computer Science, vol. 2422, Springer-Verlag, 2002, pp. 99–116.

[49] LUTZ SCHRÖDER, *Henkin models for the partial λ-calculus*, manuscript, `http://www.informatik.uni-bremen.de/~lschrode/hascasl/henkin.ps`, 2002.

[50] KURT SCHÜTTE, *Semantical and syntactical properties of simple type theory*, this JOURNAL, vol. 25 (1960), pp. 305–326.

[51] JÖRG SIEKMANN, CHRISTOPH BENZMÜLLER, et al., *Proof development with OMEGA*, ***Proceedings of the 18th international conference on automated deduction*** (Copenhagen, Denmark) (Andrei Voronkov, editor), Lecture Notes in Artificial Intelligence, vol. 2392, Springer-Verlag, 2002, pp. 144–149.

[52] RAYMOND M. SMULLYAN, *A unifying principle for quantification theory*, ***Proceedings of the National Academy of Sciences***, vol. 49 (1963), pp. 828–832.

[53] ———, ***First-order logic***, Springer-Verlag, 1968.

[54] MOTO-O TAKAHASHI, *Cut-elimination in simple type theory with extensionality*, ***Journal of the Mathematical Society of Japan***, vol. 19 (1967), pp. 399–410.

[55] GAISI TAKEUTI, ***Proof theory***, North-Holland, 1987.

[56] R. TOMASON, *A model theory for propositional attitudes*, ***Linguistics and Philosophy***, vol. 4 (1980), pp. 47–70.

[57] JEAN VAN HEIJENOORT, ***From Frege to Gödel: a source book in mathematical logic 1879–1931,*** 3rd printing, 1997 ed., Source books in the history of the sciences series, Harvard University Press, Cambridge, MA, 1967.

[58] DAVID A. WOLFRAM, *A semantics for λ-PROLOG*, ***Theoretical Computer Science***, vol. 136 (1994), no. 1, pp. 277–289.

DEPARTMENT OF COMPUTER SCIENCE
SAARLAND UNIVERSITY
SAARBRÜCKEN, GERMANY
E-mail: chris@ags.uni-sb.de
URL: http://www.ags.uni-sb.de/~chris

DEPARTMENT OF MATHEMATICS
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, PA 15213, USA
E-mail: cebrown@andrew.cmu.edu
URL: http://www.andrew.cmu.edu/~cebrown/

SCHOOL OF ENGINEERING AND SCIENCES
INTERNATIONAL UNIVERSITY BREMEN
BREMEN, GERMANY
and
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIVERSITY
PITTSBURGH, USA
E-mail: m.kohlhase@iu-bremen.de
URL: http://www.cs.cmu.edu/~kohlhase